

# RX23W Group

# BLE Module Firmware Integration Technology

## Introduction

This application note describes the BLE module which uses Firmware Integration Technology (FIT). This module controls Bluetooth<sup>®</sup> Low Energy communication. In this document, this module is referred to as the BLE FIT module.

## **Target Device**

RX23W Group

## **Related Documents**

- Bluetooth Core Specification (<u>https://www.bluetooth.com</u>)
- RX23W Group User's Manual: Hardware (R01UH0823)
- Firmware Integration Technology User's Manual (R01AN1833)
- RX Family Board Support Package Module Using Firmware Integration Technology (R01AN1685)
- Adding Firmware Integration Technology Modules to Projects (R01AN1723)
- Adding Firmware Integration Technology Modules to CS+ Projects (R01AN1826)
- RX Family Renesas FreeRTOS(R01AN4307)
- Renesas e2 studio Smart Configurator User Guide (R20AN0451)
- RX23W Group Tuning procedure of Bluetooth dedicated clock frequency (R01AN4762)
- Bluetooth Low Energy Profile Developer's Guide (R01AN6459)
- Bluetooth Low Energy Protocol Stack Basic Package: User's Manual (R01UW0205)
- RX23W Group Bluetooth Low Energy Application Developer's Guide(R01AN5504)
- Bluetooth Low Energy MCU Bluetooth Test Tool Suite operating instructions(R01AN4554)

The *Bluetooth*<sup>®</sup> word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by Renesas Electronics Corporation is under license. Other trademarks and registered trademarks are the property of their respective owners.



# Contents

1.	Overview	5
1.1	How to Use	5
1.2	Software Structure	6
1.3	Directory / File structure	7
2	APLInformation	R
2. 21		 Q
2.1	Software Requirements	o
2.2		10
2.0	Connection parameters Limitations in Perinheral role	10
2.3.1	Supported Toolchain	11
2.5	Header Files	11
2.0	Integer Types	11
2.0	Compile Configuration	12
2.8	Feature of Bluetooth LE Protocol Stack	19
2.9		20
2.9.1	Abstraction API	20
2.9.2	Software timer	20
2.9.3	Security data management	20
2.9.4	Profile common	20
2.9.5	Logger	20
2.9.6	Command Line	21
2.9.7	LED and Switch control	22
2.9.8	BLE task control	23
2.10	Device-specific data	23
2.11	Code Size	25
2.12	Adding the FIT Module to Your Project	26
3.	R_BLE API functions	27
4.	BLE FIT module project	28
4.1	Create a new project	28
4.2	Confirm BSP version	32
4.3	Clock configuration	33
4.4	Adding FIT modules	34
4.5	Configuration options	37
4.5.1	BSP(r_bsp)	37
4.5.2	Command Line Interface	38
4.5.3	Security Data Management/Device-specific data (in data flash)	40
4.5.4	LED and Switch control	41



# RX23W Group

4.5.5	Renesas FreeRTOS	43
4.6	Configuration after code generation	46
4.6.1	Changing CMT (if the CMT FIT module version is prior to 4.50)	46
4.6.2	LED and Switch control for customer board	47
4.6.3	Add application code	49
4.6.4	Profile Common code generation	50
4.7	Linker configurations	51
4.7.1	Bluetooth LE Protocol Stack program section separation	51
4.7.2	Bluetooth LE Protocol Stack Library Configuration	54
4.8	Debug configurations	56
4.8.1	Flash ID Code	56
4.8.2	Power	57
4.9	IAR development environments	58
4.9.1	Create a project	58
4.9.2	Add iodefines.h for IAR	58
4.9.3	Project conversion	61
4.9.4	Project options configuration	63
4.9.5	Exclude some directories from the build target	69
4.9.6	Build the project and download the firmware	70
5	Application Guide	71
6.	Renesas FreeRTOS BLE task	.72
6.   6.1	Call available API	.72 72
6. 6.1 6.2	Renesas FreeRTOS BLE task Call available API BLE task generation	. 72 72 73
6. 6.1 6.2 6.2.1	Renesas FreeRTOS BLE task Call available API BLE task generation Generation as main task	72 72 73 73
6. 6.1 6.2 6.2.1 6.2.2	Renesas FreeRTOS BLE task Call available API BLE task generation Generation as main task Generation as subtask	72 72 73 73 75
6. 6.1 6.2 6.2.1 6.2.2 6.3	Renesas FreeRTOS BLE task Call available API BLE task generation Generation as main task Generation as subtask BLE GATT App task generation	72 72 73 73 75 77
6. 6.1 6.2 6.2.1 6.2.2 6.3 6.4	Renesas FreeRTOS BLE task Call available API BLE task generation Generation as main task Generation as subtask BLE GATT App task generation Wake up BLE task from another task	72 72 73 73 75 75 77
6. 6.1 6.2 6.2.1 6.2.2 6.3 6.4 6.5	Renesas FreeRTOS BLE task Call available API BLE task generation Generation as main task Generation as subtask BLE GATT App task generation Wake up BLE task from another task Wake up BLE task from interrupt other than RF	72 73 73 75 75 77 79 80
6. 6.1 6.2 6.2.1 6.2.2 6.3 6.4 6.5 6.5.1	Renesas FreeRTOS BLE task Call available API BLE task generation Generation as main task Generation as subtask BLE GATT App task generation Wake up BLE task from another task Wake up BLE task from interrupt other than RF Register task waked up by command input	72 73 73 75 77 79 80 80
6. 6.1 6.2 6.2.1 6.2.2 6.3 6.4 6.5 6.5.1 6.5.2	Renesas FreeRTOS BLE task Call available API BLE task generation Generation as main task Generation as subtask BLE GATT App task generation Wake up BLE task from another task Wake up BLE task from interrupt other than RF Register task waked up by command input Wake BLE task up using LED & SW	72 72 73 73 75 77 79 80 80
6. 6.1 6.2 6.2.1 6.2.2 6.3 6.4 6.5 6.5.1 6.5.2 7.	Renesas FreeRTOS BLE task	72 72 73 73 75 77 79 80 80 80
6. 6.1 6.2 6.2.1 6.2.2 6.3 6.4 6.5 6.5.1 6.5.2 7. 7.	Renesas FreeRTOS BLE task Call available API BLE task generation Generation as main task Generation as subtask BLE GATT App task generation Wake up BLE task from another task Wake up BLE task from interrupt other than RF Register task waked up by command input Wake BLE task up using LED & SW Tools	72 72 73 73 75 77 79 80 80 80 81
6. 6.1 6.2 6.2.1 6.2.2 6.3 6.4 6.5 6.5.1 6.5.2 7. 7.1 7.2	Renesas FreeRTOS BLE task	72 72 73 73 77 79 80 80 80 81 81
6. 6.1 6.2 6.2.1 6.2.2 6.3 6.4 6.5 6.5.1 6.5.2 7. 7.1 7.2 8	Renesas FreeRTOS BLE task Call available API BLE task generation Generation as main task Generation as subtask BLE GATT App task generation Wake up BLE task from another task Wake up BLE task from interrupt other than RF Register task waked up by command input Wake BLE task up using LED & SW Tools BDAddrWriter CLVALTune	72 72 73 73 77 79 80 80 80 81 81 81
6. 6.1 6.2 6.2.1 6.2.2 6.3 6.4 6.5 6.5.1 6.5.2 7.1 7.2 8. 8.	Renesas FreeRTOS BLE task	72 72 73 73 75 77 79 80 80 81 81 81 81
6. 6.1 6.2 6.2.1 6.2.2 6.3 6.4 6.5 6.5.1 6.5.2 7. 7.1 7.2 8. 8.1 8.2	Renesas FreeRTOS BLE task         Call available API         BLE task generation         Generation as main task         Generation as subtask         BLE GATT App task generation         Wake up BLE task from another task.         Wake up BLE task from interrupt other than RF         Register task waked up by command input         Wake BLE task up using LED & SW         Tools         BDAddrWriter         CLVALTune         Demo Projects         GATT Cliept demo project         GATT Cliept demo project	72 73 73 73 77 79 80 80 80 81 81 81 81 81
6. 6.1 6.2 6.2.1 6.2.2 6.3 6.4 6.5 6.5.1 6.5.2 7. 7.1 7.2 8. 8.1 8.2 8.2 8.2 8.2	Renesas FreeRTOS BLE task	72 72 73 73 75 77 79 80 80 80 81 81 81 81 81
6. 6.1 6.2 6.2.1 6.2.2 6.3 6.4 6.5 6.5.1 6.5.2 7.1 7.2 8. 8.1 8.2 8.3 8.4	Renesas FreeRTOS BLE task	72 72 73 75 77 79 80 80 80 81 81 81 81 81
6. 6.1 6.2 6.2.1 6.2.2 6.3 6.4 6.5 6.5.1 6.5.2 7. 7.1 7.2 8. 8.1 8.2 8.3 8.4 8.4 8.5	Renesas FreeRTOS BLE task	72 72 73 73 77 79 80 80 80 80 81 81 81 81



8.6	Importing the demo project	93	
9.	Appendices	95	
9.1	Confirmed Operation Environment	95	
9.2	Troubleshooting	96	
Rev	ision History	97	
General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products 102			
Noti	ce1	03	



## 1. Overview

The BLE FIT module provides Bluetooth Low Energy (hereinafter referred to as "Bluetooth LE") features which are compliant with Bluetooth Core Specification version 5.0. The BLE FIT module supports the following features.

Bluetooth 5.0 features:

- LE 2M PHY
- LE Coded PHY
- LE Advertising Extensions
- LE Channel Selection Algorithm #2
- High Duty Cycle Non-Connectable Advertising

Bluetooth 4.2 features:

- LE Secure Connections
- Link Layer Privacy
- Link Layer Extended Scanner Filter policies
- LE Data Packet Length Extension

Bluetooth 4.1 features:

- LE L2CAP Connection Oriented Channel Support
- Low Duty Cycle Directed Advertising
- 32-bit UUID Support in LE
- LE Link Layer Topology
- LE Ping

## 1.1 How to Use

The BLE FIT module can be implemented in a project and used as the API. Refer to Section 2.12 - Adding the FIT Module to Your Project - for details on implementing the module to the project.



## 1.2 Software Structure

Figure 1-1 shows the software structure of the BLE FIT module.



Figure 1-1 : Software structure

The BLE FIT module consists of the Bluetooth LE Protocol Stack and app\_lib.

The Bluetooth LE Application uses the Bluetooth LE functions via the R\_BLE API provided by the Bluetooth LE Protocol Stack.

The app\_lib includes optional functions available for the Bluetooth LE Application. The abstraction API of the R\_BLE API also is an interface for the Bluetooth LE functions.

The QE for BLE generates the source codes (Bluetooth LE base skeleton program) as a base for the Bluetooth LE Application and the Bluetooth LE Profiles.

The BLE FIT v2.50 or later doesn't include Profile Common in app\_lib. Profile Comon is provided by QE for BLE V1.6.0 or later.

Renesas recommends using the QE for BLE for the development of the Bluetooth LE Application.



## 1.3 Directory / File structure

Table 1.1 shows the directory / file structure of the BLE FIT module.

Directory /File s	structure				Description		
		en	r01an4860ej0250-rx23w-ble.pdf		Application Note(English)		
	doc	jp	r01an4860jj0	250-rx23w-ble.pdf	Application Note(Japanese)		
		r_ble_api_spec.ch	im		R_BLE API document		
		ble_fit_lib_selecto	r.bat	Library selector bat file			
		ble_fit_lib_selecto	r.sh	Library selector shell script			
		lib_ble_ps_ccrx_a	lib	Bluetooth LE Protocol Stack(ALL Features)			
	lib	lib_ble_ps_ccrx_b	.lib	Bluetooth LE Protocol Stack(Balance)			
		lib_ble_ps_ccrx_c	lib	Bluetooth LE Protocol Stack(Compact)			
		lib_ble_ps_hci_cc	rx_a.lib		HCI mode library (ALL Features)		
		lib_ble_ps_hci_cc	rx_b.lib		HCI mode library (Balance)		
		lib_ble_ps_hci_cc	rx_c.lib	HCI mode library (Compact)			
	ref r_ble_rx23w_config_reference.h				Configuration reference file		
r ble rx23w			abs		Abstraction API(GAP)		
		app_lib	board		Control the LED and SW on the board		
			cli		Command Line(input/output)		
			cmd		Command Line(command)		
			logger		Logger		
			rtos		BLE task control		
	src		sec_data		Security data management		
			timer		Software timer		
		platform	driver	dataflash	Data Flash driver for BLE		
			r_ble_pf_config_private.h		BLE configuration control		
			r_ble_pf_configs.c				
			r_ble_pf_functions.c		RF driver dependent on platform		
		r_ble_pf_lowpower.c			Low power consumption program		
	r_ble_rx2	3w_if.h			BLE interface file		
	readme.t	xt			readme		
r_config	r_ble_rx2	3w_config.h			Configuration option file		

Table 1.1	: Directory /	<sup>/</sup> File structure
-----------	---------------	-----------------------------

The BLE FIT v2.50 or later doesn't include Profile Common(profile\_cmn, discovery).



## 2. API Information

The BLE FIT module API follows the Renesas API naming standards.

## 2.1 Hardware Requirements

The Bluetooth LE Protocol Stack in the BLE FIT always uses the RX23W hardware resources in Table **2.1**. Therefore, user program cannot use these hardware resources.

Table 2.1 : RX23W Hardware F	Resources used by the BLE FIT
------------------------------	-------------------------------

H/W Resources
BLE
CMT: CMT2, CMT3
Interrupt: BLEIRQ, BLE.ERI/RXI/TXI/TEI

The priority level of BLEIRQ, BLE.ERI/RXI/TXI/TEI is 14.

The BLE FIT options use the RX23W hardware resources in Table 2.2. Therefore, user program cannot use these hardware resources which are selected by the options. For more information about each option, refer to "2.9 app\_lib".

H/W Resources	Option	Note		
CMT: CMT0 or CMT1	Software timer	Bare metal: The CMT FIT allocates dynamically CMT0 or CMT1 for the software timer.		
		FreeRTOS: The software timer uses the FreeRTOS timer API.		
	Command Line	SCIn.ERI/RXI/TXI/TEI are used. (n is the channel number used by Command Line.)		
Interrupt	LED and Switch control	LED and Switch control uses interrupts for the board switches. These interrupts are configured on the Smart Configurator. For more information about the interrupts, refer to Table 4.1.		
Data Flash	Device-specific data	The Data Flash area used by each option is selected on the Smart Configurator. For more information about these Data Flash areas, refer to		
	Security data management	"4.5.3 Security Data Management/Device-specific data (in data flash)".		
SCI	Command Line	The SCI channel used by the Command Line is configured on the Smart Configurator. For more information about the channel, refer to "4.5.2 Command Line Interface ".		
Port	LED and Switch control	The ports used by the LED and Switch control are configured on the Smart Configurator. For more information about the port, refer to Table 4.1.		

Table 2.2 : RX23W Hardware Resources used by the BLE FIT options



## 2.2 Software Requirements

The BLE FIT module is dependent upon the following FIT modules:

- Renesas Board Support Package (r\_bsp)
   If you use the BLE FIT module version 1.01 or later, use BSP version 5.40 or later.
   If you select R5F523W8CxLN or R5F523W8DxLN, use BSP version 5.62 or later.
- CMT (r\_cmt\_rx, version 4.10 later) The Bluetooth LE Protocol Stack uses CMT2, CMT3. If the CMT FIT module version is prior to 4.50, modify the CMT code (Refer "4.6.1 Changing CMT"). If the version is 4.50 or later, you do not need change the CMT code. When the software timer(app\_lib/timer) is in use, it uses another one CMT channel.
- LPC (r\_lpc\_rx)

Also, the following FIT modules are required by the configuration option of the BLE FIT module.

Command Line Interface (BLE\_CFG\_CMD\_LINE\_EN=1):

- SCI (r\_sci\_rx)
- byte queues/circular buffers (r\_byteq\_rx)

Security Data Management (BLE\_CFG\_EN\_SEC\_DATA=1) or device-specific data (data area) (BLE\_CFG\_DEV\_DATA\_DF\_BLOCK=0 – 7):

• Data Flash (r\_flash\_rx)

LED and Switch control (BLE\_CFG\_BOARD\_LED\_SW\_EN=1):

- GPIO (r\_gpio\_rx) If you select R5F523W8CxLN or R5F523W8DxLN, use GPIO version 3.80 or later.
- IRQ (r\_irq\_rx)



## 2.3 Limitations

#### 2.3.1 Connection parameters Limitations in Peripheral role

• BLE FIT v2.40 or earlier version have the following limitation.

If the Connection Interval is less than 66.25ms when the RX23W connects as a peripheral role, it may not be possible to transition to RF sleep mode between disconnection due to Supervision Timeout and reconnection.

To avoid this situation, set the Supervision Timeout to less than 300 times the Connection Interval.

Example: Connection Interval: 20ms, set Central's connection parameters so that Supervision Timeout: 20ms \* 300 = 6.0 seconds or less.

If the Supervision Timeout value is set from Central that exceeds the above conditions, update the connection parameter from the peripheral side using R\_BLE\_GAP\_UpdConn() API.

#### • BLE FIT v2.50 or later version have the following limitation.

When the RX23W is connected as a peripheral role and the Connection Interval is set to 66.25ms or less, the central role may disconnect earlier than the Supervision Timeout set by the central.

The maximum amount of time that a connection can be maintained depends on the central Sleep Clock Accuracy (SCA) notified in the connection request.

Table 2.3 shows the maximum amount of time that a connection can be maintained depending on the set Connection Interval and central SCA value.

Connection	Central Sleep Clock Accuracy							
Interval	0	1	2	3	4	5	6	7
	(500ppm)	(250ppm)	(150ppm)	(100ppm)	(75ppm)	(50ppm)	(30ppm)	(20ppm)
7.5 ms	3390 ms	5670 ms	5670 ms	8520 ms				
15.0 ms	8160 ms	12870 ms	15000 ms	18030 ms	22560 ms	22560 ms	22560 ms	22560 ms
20.0 ms	11320 ms	17000 ms	21260 ms	24300 ms	28360 ms	28360 ms	32000 ms	32000 ms
30.0 ms	18360 ms	26970 ms	31980 ms					
40.0 ms	24600 ms	32000 ms	32000 ms	32000 ms	32000 ms	32000 ms	32000 ms	32000 ms
50.0 ms	30800 ms	32000 ms	32000 ms	32000 ms	32000 ms	32000 ms	32000 ms	32000 ms
60.0 ms	31980 ms	31980 ms	31980 ms	31980 ms	31980 ms	31980 ms	31980 ms	31980 ms
66.25 ms	31800 ms	31800 ms	31800 ms	31800 ms	31800 ms	31800 ms	31800 ms	31800 ms
67.5 ms to	no limit	no limit	no limit	no limit	no limit	no limit	no limit	no limit
4 sec								

Table 2.3 : Maximum time of connection maintenance by Connection Interval and central SCA value

In the RX23W peripheral operation, a connection with Supervision Timeout that exceeds these maximum amounts of time is accepted, but when the central packet cannot be received reaches the time, it will be disconnected, and the disconnection event (BLE\_GAP\_EVENT\_DISCONN\_IND) will be called back to the application.

If Supervision Timeout is less than the maximum, the connection will be disconnected when the time specified by Supervision Timeout is reached.

This limitation does not affect the central operation of RX23W.



## 2.4 Supported Toolchain

The BLE FIT module has been confirmed to work with the toolchain listed in "9.1 Confirmed Operation Environment".

## 2.5 Header Files

All API calls and their supporting interface definitions are located in r\_ble\_rx23w\_if.h.

## 2.6 Integer Types

This project uses ANSI C99 "Exact width integer types" in order to make the code clearer and more portable. These types are defined in stdint.h.



## 2.7 Compile Configuration

The configuration options of the BLE FIT module are located in r\_ble\_rx23w\_config.h. The options can be configured in Software component configuration screen when using Smart Configurator. The changed options are automatically reflected when adding the BLE FIT module to the project. The option names and setting values are listed in the table below:

Configuration options (r_ble_rx2'3w_config.h)	
BLE_CFG_LIB_TYPE Default: "0"	Type of the Bluetooth LE Protocol Stack. Select one of the followings. 0: All features 1: Balance 2: Compact
	Refer to "2.8 Feature of Bluetooth LE Protocol Stack" for details.
BLE_CFG_RF_DBG_PUB_ADDR Default: "{0xFF,0xFF,0xFF,0x50,0x90,0x74}"	Initial Public Address. If the public addresses in the Code Flash and the Data Flash are all 0x00 or 0xFF, the device adopts this public address. If all 0x00 or 0xFF is set, the device uses 74:90:50:FF:FF:FF as public address.
BLE_CFG_RF_DBG_RAND_ADDR Default: "{0xFF,0xFF,0xFF,0xFF,0xFF,0xFF}"	Initial Static Address. If the static addresses in the Code Flash and the Data Flash are all 0x00 or 0xFF, the device adopts this static address. If all 0x00 or 0xFF is set, the device uses the value generated with the device specific value the static address.
BLE_CFG_RF_CONN_MAX Default: "7"	Maximum entry number of the remote device connection information management table.
	Specifically, it is either larger one of the maximum simultaneous connection numbers and the maximum stored bonding information number.
	The heap memory size used by the Bluetooth LE Protocol Stack increases and decreases due to this value. Range: 1 to 7
BLE_CFG_RF_CONN_DATA_MAX Default: "251"	Maximum packet data length (bytes). Range: 27 to 251
BLE_CFG_RF_ADV_DATA_MAX Default: "1650"	Maximum advertising data length (bytes). Range: 31 to 1650
	The maximum advertising data length of the Bluetooth LE Protocol Stack libraries other than "All features" is fixed to 31bytes.



Configuration options (r_ble_rx23w_config.h)	
BLE_CFG_RF_ADV_SET_MAX Default: "4"	Maximum number of the advertising set. Range: 1 to 4
	The number of the advertising set of the Bluetooth LE Protocol Stack libraries other than "All features" is fixed to one.
BLE_CFG_RF_SYNC_SET_MAX Default: "2"	Maximum number of periodic sync set. Range: 1 to 2
	If the Bluetooth LE Protocol Stack library is other than "All features", this option is not used.
BLE_CFG_EVENT_NOTIFY_CONN_START Default: "0"	Enable or disable start interrupt notification of a connection complete event.
	0: Disable 1: Enable
	Because the start notification is triggered by the interrupt, it occurs after the actual RF event.
BLE_CFG_EVENT_NOTIFY_CONN_CLOSE Default: "0"	Enable or disable end interrupt notification of a connection complete event.
	0: Disable 1: Enable
BLE_CFG_EVENT_NOTIFY_ADV_START Default: "0"	Enable or disable the advertising event start interrupt notification. 0: Disable 1: Enable
	<ul> <li>The notification occurs at the following timings.</li> <li>Start Primary Advertising channel.</li> <li>Start Secondary Advertising Channel</li> <li>Start Periodic Advertising. (When the Extended Advertising is enabled.)</li> </ul>
	Because the start notification is triggered by the interrupt, it occurs after the actual RF event.
BLE_CFG_EVENT_NOTIFY_ADV_CLOSE Default: "0"	Enable or disable the advertising event complete interrupt notification. 0: Disable 1: Enable
	The notification occurs at the following timings. - Complete Primary Advertising channel. - Complete Secondary Advertising Channel - Complete Periodic Advertising. (When the Extended Advertising is enabled.)
	If the advertising is terminated by a command, the notification doesn't occur.



Configuration options (r_ble_rx23w_config.h)	
BLE_CFG_EVENT_NOTIFY_SCAN_START Default: "0"	Enable or disable the scan start interrupt notification. 0: Disable 1: Enable
	If the scan interval is equal to the scan window, this notification doesn't occur. Because the start notification is triggered by the interrupt, it occurs after the actual RF event.
BLE_CFG_EVENT_NOTIFY_SCAN_CLOSE	Enable or disable the scan complete interrupt notification
	0: Disable 1: Enable
	If the scan interval is equal to the scan window, this notification doesn't occur. If the scan is terminated by a command, the notification doesn't occur.
BLE_CFG_EVENT_NOTIFY_INIT_START Default: "0"	Enable or disable the notification that the scan start interrupt has occurred in sending a connection request.
	0: Disable 1: Enable
	If the scan interval is equal to the scan window, this notification doesn't occur. Because the start notification is triggered by the interrupt, it occurs after the actual RF event.
BLE_CFG_EVENT_NOTIFY_INIT_CLOSE Default: "0"	Enable or disable the notification that the scan complete interrupt has occurred in sending a connection request.
	0: Disable 1: Enable
	If the scan interval is equal to the scan window, this notification doesn't occur. If the connection request is terminated by a command, the notification doesn't occur.
BLE_CFG_EVENT_NOTIFY_DS_START Default: "0"	Enable or disable the RF_DEEP_SLEEP start notification. 0: Disable 1: Enable
BLE_CFG_EVENT_NOTIFY_DS_WAKEUP Default: "0"	Enable or disable the RF_DEEP_SLEEP wakeup notification.
	0: Disable 1: Enable



Configuration options (r_ble_rx23w_config.h)	
BLE_CFG_RF_CLVAL Default: "6"	Adjustment value of the 32MHz crystal oscillator. Set this option according to the board environment. Range: 0 to 15
	Refer to"RX23W Group Tuning procedure of Bluetooth dedicated clock frequency(R01AN4762)" for details. If you select R5F523W8CxLN or R5F523W8DxLN, set "7".
BLE_CFG_RF_DDC_EN Default: "0"	Enable or disable the DC-DC on the RF.
	0: Disable 1: Enable
	If you use Target Board, set "0".
BLE_CFG_RF_EXT32K_EN Default: "0"	Slow clock source to the RF. Range: 0 to 1
	0: RF_LOCO 1: External 32.768kHz
	If this option is set to 1, the sub clock is required to be enabled in the Smart Configurator clock configuration screen ( <b>Figure 4-11</b> ).
BLE_CFG_RF_MCU_CLKOUT_PORT Default: "0"	Port of the MCU CLKOUT. Range: 0 to 1
	0: PE3 1: PE4
	If the BLE_CFG_RF_EXT32K_EN option is 0, this option is ignored.
BLE_CFG_RF_MCU_CLKOUT_FREQ Default: "0"	Output frequency from the MCU CLKOUT. Range: 0 to 1
	0: MCU CLKOUT frequency 32.768kHz 1: MCU CLKOUT frequency 16.384kHz
	If the BLE_CFG_RF_EXT32K_EN option is 0, this option is ignored.
BLE_CFG_RF_SCA Default: "250"	Sleep Clock Accuracy (SCA) for the RF slow clock. Range: 0 to 500
	If the BLE_CFG_RF_EXT32K_EN option is 0, the SCA is fixed to more than 250ppm and this option is ignored.
BLE_CFG_RF_MAX_TX_POW Default: "1"	Maximum transmit power configuration. Range: 0 to 1
	0: max +0dBm 1: max +4dBm



Configuration options (r_ble_rx23w_config.h)	
BLE_CFG_RF_DEF_TX_POW Default: "0"	Default transmit power level. Range: 0 to 2 This option depends on the BLE_CFG_RF_MAX_TX_POW option.
	If the BLE_CFG_RF_MAX_TX_POW option is 0(0dBm), the BLE_CFG_RF_DEF_TX_POW is as follows. 0(High): 0dBm 1(Mid): 0dBm 2(Low):-18dBm
	If the BLE_CFG_RF_MAX_TX_POW option is 1(+4dBm), the BLE_CFG_RF_DEF_TX_POW is as follows. 0(High): +4dBm 1(Mid): 0dBm 2(Low): -20dBm
BLE_CFG_RF_CLKOUT_EN Default: "0"	CLKOUT_RF output. Select one of the followings.
	0: No output 5: 4MHz output 6: 2MHz output 7: 1MHz output
BLE_CFG_RF_DEEP_SLEEP_EN Default: "1"	Enable or disable the RF Deep Sleep. 0: Disable 1: Enable
BLE_CFG_MCU_MAIN_CLK_KHZ Default: "4000"	MCU main clock frequency (kHz). This option needs to be configured according to the board environment. Set the clock frequency configured in the Smart Configurator clock configuration.
	If the HOCO is used, this option is ignored. If the Main Clock is used, set a value within the range between 1000 and 20000. If the PLL Circuit is used, set a value within the range between 4000 and 20000.
	Set the frequency input in Smart Configurator Clock configuration.
BLE_CFG_DEV_DATA_CF_BLOCK Default: "16"	The Code Flash (ROM) block stored the device specific data. Range: -1 to 255 If this option is set to -1, the device specific data in the Code Flash isn't used. The blocks from "0" to "15" are the Start-Up Program Protection block. If the Start-Up Program Protection is used, don't use the blocks from "0" to "15".
BLE_CFG_DEV_DATA_DF_BLOCK Default: "-1"	The E2 Data Flash block stored the device specific data. Range: -1 to 7 If this option is set to -1, the device specific data in the E2 Data Flash isn't used. Set the block number different from the one specified by BLE_CFG_SECD_DATA_DF_BLOCK.



Configuration options (r_ble_rx23w_config.h)	
BLE_CFG_GATT_MTU_SIZE Default: "247"	The MTU size (bytes) for the GATT communication. Range: 23 to 247
	The Profile Common responds to the MTU request from the remote device to BLE_CFG_GATT_MTU_SIZE.
BLE_CFG_NUM_BOND Default: "7"	Maximum number of the bonding information stored in the Data Flash by the security data management. Range: 1 to 7
	If you change this value for debug and so on after code generation and build, repairing may fail because bonding information managed by the security data management function before the change remains in the Data Flash area. To avoid this, delete the previous bonding information by R_BLE_GAP_DeleteBondInfo() or "gap auth del remote all" command.
BLE_CFG_EN_SEC_DATA Default: "0"	Enable or disable the security data management. The bonding information is stored in the Data Flash block specified by "BLE_CFG_SECD_DATA_DF_BLOCK" by this option.
	0: Disable 1: Enable
	If this option is enabled, add the Data Flash FIT module.
BLE_CFG_SECD_DATA_DF_BLOCK Default: "0"	The Data Flash block for the security data management to store the bonding information. Range: 0 to 7
	Set the block number different from the one specified by BLE_CFG_DEV_DATA_DF_BLOCK.
BLE_CFG_CMD_LINE_EN Default: "0"	Enable or disable the command line function.
	0: Disable 1: Enable
	If this option is enabled, add the SCI FIT module.
BLE_CFG_CMD_LINE_CH Default: "1"	SCI Channel for the command line function. Enable the SCI channel for the command line in the SCI FIT module configuration.
	If the BLE_CFG_CMD_LINE_EN is 0, this option is ignored.
BLE_CFG_BOARD_LED_SW_EN Default: "0"	Enable or disable support the board LED & Switch control.
	0: Disable 1: Enable
	If the option is enabled, add the IRQ FIT module and the GPIO FIT module.



Configuration options (r_ble_rx23w_config.h)	
BLE_CFG_BOARD_TYPE Default: "0"	Board type. Range: 0 to 3 0: Customer board 1: Target Board 2: RSSK 3: Evaluation board
BLE_CFG_LOG_LEVEL Default: "3"	Log level. Range: 0 to 3 0: disable 1: Error 2: Error & Warning 3: Error & Warning & Debug
BLE_CFG_ABS_API_EN Default: "1"	Enable or disable support the Abstraction API. 0: Disable 1: Enable
BLE_CFG_SOFT_TIMER_EN Default: "1"	Enable or disable support the software time in app_lib. 0: Disable 1: Enable If you use the Abstraction API, enable the software timer.
BLE_CFG_MCU_LPC_EN Default: "1"	Enable or disable support the MCU low power consumption control. 0: Disable 1: Enable
BLE_CFG_HCI_MODE_EN Default: "0"	Select start in HCI mode or not. 0: Not start in HCI mode 1: Start in HCI mode



## 2.8 Feature of Bluetooth LE Protocol Stack

The features of the Bluetooth LE Protocol Stack depends on the BLE\_CFG\_LIB\_TYPE option (r\_ble\_rx23w\_config.h). Table 2.5 shows the features of each type of the Bluetooth LE Protocol Stack.

See "Bluetooth Low Energy Protocol Stack Basic Package: User's Manual (R01UW0205) 3.1.2 Library of Bluetooth LE Protocol Stack" for the details of each feature.

	Bluetooth LE Protocol Stack		
Bluetooth LE Feature	All features	Balance	Compact
LE 2M PHY	Yes	Yes	No
LE Coded PHY	Yes	Yes	No
LE Advertising Extensions	Yes	No	No
LE Channel Selection Algorithm #2	Yes	Yes	No
High Duty Cycle Non-Connectable Advertising	Yes	Yes	Yes
LE Secure Connections	Yes	Yes	Yes
Link Layer privacy	Yes	Yes	Yes
Link Layer Extended Scanner Filter policies	Yes	Yes	No
LE Data Packet Length Extension	Yes	Yes	Yes
LE L2CAP Connection Oriented Channel Support	Yes	No	No
Low Duty Cycle Directed Advertising	Yes	Yes	Yes
LE Link Layer Topology	Yes	Yes	No
LE Ping	Yes	Yes	Yes
GAP Role	Central Peripheral Observer Broadcaster	Central Peripheral Observer Broadcaster	Peripheral Broadcaster
GATT Role	Sever Client	Sever Client	Sever Client
32-bit UUID Support in LE	Yes	Yes	Yes

|--|



## 2.9 app\_lib

The app\_lib provides optional functions available for the Bluetooth LE Application. The functions included in the app\_lib are as follows.

## 2.9.1 Abstraction API

Abstraction API simplifies the procedure used with the Bluetooth LE Protocol Stack. Refer to "R\_BLE API document (r\_ble\_api\_spec.chm)" for details. If the Abstraction API is used, set the BLE\_CFG\_ABS\_API\_EN option and the BLE\_CFG\_SOFT\_TIMER\_EN option to "1". Do not change the Abstraction API codes.

#### 2.9.2 Software timer

The bare metal version and the Renesas FreeRTOS version of software timer vary the implementation.

- Bare metal Software timer uses the CMT. If the software timer is used, add the CMT FIT module to the application. The CMT channel is dynamically allocated by the CMT FIT module.
- Renesas FreeRTOS Software timer uses the FreeRTOS Timer API.

If the Software timer is used, set the BLE\_CFG\_SOFT\_TIMER\_EN option to "1".

#### 2.9.3 Security data management

Security data management provides the interfaces which are used for storing or getting the bonding information in the Data Flash. If the security data management is used, add the Flash FIT module and set the BLE\_CFG\_EN\_SEC\_DATA option to "1". The BLE\_CFG\_SECD\_DATA\_DF\_BLOCK option indicates the Data Flash block for the security data management.

#### 2.9.4 Profile common

This function provides the common interfaces in the Bluetooth LE Profiles. The interfaces are called by the code generated by the QE for BLE. Refer to "Bluetooth Low Energy Profile Developer's Guide(R01AN6459)" for the details of the profile development and the profile common. The BLE FIT v2.50 or later doesn't include Profile common. Profile common is provided by QE for BLE (V1.6.0).

## 2.9.5 Logger

This function outputs the message. The BLE\_CFG\_LOG\_LEVEL option sets the output level. If you use CCRX C++, the logger function is not supported.



#### 2.9.6 Command Line

Command Line controls the Bluetooth LE functions via a serial interface. This function uses the SCI.

#### 1) Component

To configure the channel for command line, follow these steps:

- 1. Add the SCI FIT module in the Smart Configurator and select the channel for the command line.
- 2. Set the BLE\_CFG\_CMD\_LINE\_EN option to 1.
- 3. Set the BLE\_CFG\_CMD\_LINE\_CH option to the selected channel number.

#### 2) Terminal

Connect the board to a computer and launch Terminate software on the computer with the following settings.

Item	Settings
New-line (Receive)	LF
New-line (Transmit)	CR
Terminal Mode	VT100
Baud rate	115200
Data	8bit
Parity	none
Stop bits	1bit
Flow Control	none

Table 2.6 : Terminal configuration	n
------------------------------------	---



#### 3) Command

Table 2.7 presents the command supported by the command line function.

\_

Command	Sub command	Description	
	adv	Start or stop advertising.	
	scan	Start scan.	
	conn	Send a connection request.	
	disconn	Disconnect the link.	
	device	Display the addresses of the connected devices.	
gap	priv	Set the privacy feature to the local device.	
	conn_cfg	Link configuration command.	
	wl	White List operation command.	
auth	auth	Pairing or encryption command.	
	sync	Create or Terminate a periodic sync.	
	ver	Display the version information.	
	txp	Set or Get the transmit power.	
	scheme	Set the coding scheme of the Coded PHY	
vs te ac ra	test	DTM test command.	
	addr	Set or Get the local device BD_ADDR.	
	rand	Generate a random number.	
sys	stby	Operate the software standby mode.	
bla	reset	Reset the Bluetooth LE Protocol Stack.	
ble	close	Stop the Bluetooth LE Protocol Stack.	

Table 2.7	7:	Command List

If you use gap command or vs command, set "1" to BLE\_CFG\_ABS\_API\_EN with the BLE FIT module component configuration.

For more details, refer to "Bluetooth Low Energy Protocol Stack Basic Package: User's Manual (R01UW0205) 5.1 Command Line Interface".

#### 2.9.7 LED and Switch control

The LED and Switch Control feature controls LEDs and switches on board. About setting the BLE FIT and the IRQ FIT configuration options to use this feature, see "4.5.4 LED and Switch control".



#### 2.9.8 BLE task control

The BLE task control is enabled if creating a project, "FreeRTOS" is selected as the "RTOS". This feature is used if another task wakes the BLE task up.

## 2.10 Device-specific data

The Bluetooth Device Address (BD\_ADDR) can be written in the User Area (ROM) or Data Area (E2 Data Flash) of the Flash memory.

The location in which the BD\_ADDR is written is specified by BLE\_CFG\_DEV\_DATA\_CF\_BLOCK or BLE\_CFG\_DEV\_DATA\_DF\_BLOCK in the configuration option(r\_ble\_rx23w\_config.h).

If the BD\_ADDR is written in the User Area (ROM), it is necessary to specify the block which the program code doesn't use. Also, the BD\_ADDR needs to be written in the top address of the specified block.

Table 2.8 shows the format of the written data.

Offset	Size[byte]	Description
0	4	Data length after the magic number(Fixed 16)
4	4	Magic number(Fixed 0x12345678)
8	6	Public BD_ADDR
14	6	Random BD_ADDR

Table	28	•	Data	Format
rabic	2.0	٠	Data	i onnat

Data must be written in little endian for each block.

Data can be written to multiple RX23Ws continuously by using the unique code function of Renesas Flash Programmer (RFP).

A setting example of the RFP unique code file (ruc) is shown below.





The device searches for the BD\_ADDR which are not all zero and not all 0xFF in the following order.

- (1) The specified block of the Data Area (E2 Data Flash)
- (2) The specified block of the User Area (ROM)
- (3) The initial value of the firmware (BLE\_CFG\_RF\_DBG\_PUB\_ADDR or BLE\_CFG\_RF\_DBG\_RAND\_ADDR)

By default, the flash memory protection of the RX23W is disabled. If the flash memory protection is disabled, all the blocks are erased using a serial programmer connection such as the Renesas Flash Program (RFP), etc. Therefore, it is necessary to enable the flash memory protection to write a new firmware while retaining the written BD\_ADDR in the flash memory. To enable the flash memory protection, configure the r\_bsp configuration option and the ID Code following the below sections.

- 4.5.1 BSP(r\_bsp)
- 4.8.1 Flash ID Code



## 2.11 Code Size

Table 2.9 shows the code size of the Bluetooth LE Protocol Stack of the BLE FIT module. The code size is based on optimization level 2 and optimization type for size for the RXC toolchain in Section 9.1.

The ROM (code and constants) and RAM (global data) sizes are determined by the build-time configuration options set in the BLE FIT module configuration header file. The values in the table below are confirmed under the following conditions.

Dovico	Compiler	Category	Type of the	Type of the Bluetooth LE Protocol Stack							
Device	Compilei	Calegory	All Features	Balance	Compact						
RX23W	CC-RX	ROM	191K bytes	149K bytes	133K bytes						
10,2010	00100	RAM	38K bytes	23K bytes	23K bytes						
Configuration Option											
Common:											
BLE_CFG_RF_CONN_MAX 7											
BLE_CFG_	BLE_CFG_RF_CONN_DATA_MAX 251										
BLE_CFG_	NUM_BOND 7										
All Features	<u>s:</u>										
BLE_CFG_	LIB_TYPE 0										
BLE_CFG_	RF_ADV_DATA_	MAX 1650									
BLE_CFG_	RF_ADV_SET_N	IAX 4									
BLE_CFG_	RF_SYNC_SET_	MAX 2									
Balance:											
BLE_CFG_	LIB_TYPE 1										
Compact:											
BLE_CFG_	LIB_TYPE 2										

#### Table 2.9 : Code Size

NOTE:

- The app\_lib and the Bluetooth LE Profiles are not included in the above code size.
- The Bluetooth LE Protocol Stack libraries which are generated with the CCRX compiler. The IAR development environments uses the same libraries.



## 2.12 Adding the FIT Module to Your Project

This module must be added to each project in which it is used. Renesas recommends using "Smart Configurator" described in (1) or (3).

- (1) Adding the FIT module to your project using "Smart Configurator" in e2 studio. By using the "Smart Configurator" in e2 studio, the FIT module is automatically added to your project. Refer to "Renesas e2 studio Smart Configurator User Guide (R20AN0451)" for details.
- (2) Adding the FIT module to your project using "FIT Configurator" in e2 studio By using the "FIT Configurator" in e2 studio, the FIT module is automatically added to your project. Refer to "Adding Firmware Integration Technology Modules to Projects (R01AN1723)" for details.
- (3) Adding the FIT module to your project using "Smart Configurator" on CS+ By using the "Smart Configurator Standalone version" in CS+, the FIT module is automatically added to your project. Refer to "Renesas e2 studio Smart Configurator User Guide (R20AN0451)" for details.
- (4) Adding the FIT module to your project in CS+ In CS+, please manually add the FIT module to your project. Refer to "Adding Firmware Integration Technology Modules to CS+ Projects (R01AN1826)" for details.



#### 3. R\_BLE API functions

Refer to "R\_BLE API document (r\_ble\_api\_spec.chm)" for details.

If the r\_ble\_api\_spec.chm cannot be opened, right-click the file, select Properties and check "Unblock" as follows.

🚏 r_ble_api_s	pec.chm Properties	×				
General Secu	rity Details Previous Versions					
P	r_ble_api_spec.chm					
Type of file:	Compiled HTML Help file (.chm)	-				
Opens with:	👔 Microsoft® HTML Help E 🛛 Change					
Location:	C:¥Users¥a51 00807¥Downloads	_				
Size:	8.1.0 MB (8,501,807 bytes)					
Size on disk:	8.1 0 MB (8,503,296 bytes)					
Created:	Today, August 19, 2019, 8 minutes ago	_				
Modified:	Today, August 19, 2019, 8 minutes ago					
Accessed:	Today, August 19, 2019, 8 minutes ago					
Attributes:	Read-only Hidden Advanced					
Security:	This file came from another computer and might be blocked to help protect this computer.	]				
	OK Gancel Apply					

Figure 3-1 : r\_ble\_api\_spec.chm Properties



## 4. BLE FIT module project

This section describes how to create a new BLE project in the e<sup>2</sup>studio and the Smart Configurator to build a Bluetooth LE Application development environment.

## 4.1 Create a new project

Select "File"→"C/C++ Project". In "New C/C++ Project" dialog, select "Renesas RX" and "Renesas CC-RX C/C++ Executable Project" and click on the "Next" button.

e <sup>2</sup> New C/C++ P	roject			×					
Templates for N	lew C/C++ Project								
All Make Renesas Debug Renesas RX	GCC for Renesas RX C/C++ Execut A C/C++ Executable Project for Renesas for Renesas RX Toolchain. GCC for Renesas RX C/C++ Librar GCX A C/C++ Library Project for Renesas R	table P as RX us y Proje X using	roject ing the GC ct the GCC	rc					
	Renesas KA toolchain. Renesas CC-RX C/C++ Executable A C/C++ Project for Renesas RX using toolchain.	Tor Renessas NA FOORCHAIN. Renessas CC-RX C/C++ Executable Project A C/C++ Project for Renessas RX using the Renesas CCRX toolchain.							
	Renesas CC-RX C/C++ Library Pro A C/C++ Library Project for Renesas R Renesas CCRX toolchain.	ject Kusing	the						
L	л								
?	< Back Next > Finish		Cance	el					

Figure 4-1 : Templates for New C/C++ Project

Enter the project name and click on the "Next" button.

e <sup>2</sup>				_		×
New Renesas CC New Renesas CC-	-RX Executable RX Executable Proj	<b>Project</b> ect				2
Project name: sa	ample_project					
Use default le	ocation					
Location:	C:¥src¥workspa	ce¥sample_project			Browse	
	Create Direct	ory for Project				
Choose file syste	m: default 🖂					
Working sets						
Add project	to working sets				New	
Working sets:				$\sim$	Select	
?		< Back	Next >	Finish	Cance	el

Figure 4-2 : New Renesas CC-RX Executable Project



Set the "Endian" combo box to "Little" and select the RX23W type name from  $[RX200] \rightarrow [RX23W]$  in accordance with the board in the "Target Device".

e <sup>2</sup> works	pace - e² studio									-		×
File Edit	Source Refactor	Navigate Search Proje	ct Renesas View	s Run W	indow	Help				$\left[ \begin{array}{c} & \leftarrow \end{array} \right] A$	· A 🗄	
	e <sup>2</sup> New Renesas CC-F Select toolchain, de	<b>IX Executable Project</b> vice & debug settings				_	Ē	×		120	Wo	prkbench
	Toolchain Settings Language: Toolchain: Toolchain Version: RTOS:	C O C++ Renesas CCRX v2.08.00 None	~ ~ ~						view of the features			
	Device Settings Target Device: R5	Man F51101AxLM		Configura	tions • Hardy	ware Debug Cor	nfigurati	ion	tutorials			
	Endian: Litt Project Type: De	le fault	Jnlock Device	RX100 RX200 RX600	> > >	RX210 RX21A	>	~	samples			
			_	Create	> e Rele	RX220 RX230 RX231 RX23T	>	~	eW at is new			
	?	<	Back Ne	ext >	F	RX23W RX24T RX24U	> > >	F	X23W - 56pin > X23W - 85pin >	R5F523W7AxN R5F523W7BxN R5F523W8AxN R5F523W8BxN	IG IG IG IG	

Figure 4-3 : Toolchain, device & debug settings

Figure 4-4 shows the RX23W Product Part Number.



Figure 4-4 : RX23W Product Part Number

For the RX23W Product Part Number details, see "1.2 List of Products" in "RX23W Group User's Manual: Hardware (R01UH0823)".



If you use the Renesas FreeRTOS, select "FreeRTOS (kernel only)" from the "RTOS" and RTOS version.

Toolchain Settir	ngs		
Language:	● C ○ C++		
Toolchain:	Renesas CCRX	$\sim$	
Toolchain Versio	on: v2.08.00	$\sim$	
	Manage Toolcha	ins	
RTOS:	FreeRTOS (kernel only)	$\sim$	
RTOS Version:	10.4.3-rx-1.0.1	~	
	Manage RTOS Versio	ons	
Device Settings		G	onfigurations
Target Board:	Custom	~ 5	Create Hardware Debug Configuratio
	Download additional boa	rds	E1 (RX) ~
Target Device:	R5F523W8AxNG		Create Dahug Configuration
	Unlock Devic	ces	DY Circulates
Endian:	Little	$\sim$	rx simulator
Project Type:	Default	~	Create Release Configuration

Figure 4-5 : RTOS and RTOS version

To download the latest Renesas FreeRTOS, click "Mange RTOS version..." and select the latest version.

lect All
lect All
lect All
elect Al
wse

Figure 4-6 : Renesas FreeRTOS Download

If you don't need RTOS, select "None".

Click "Next" button, the following screen is "Select Coding Assistant settings" dialog is displayed.



Check the "Use Smart Configurator" option in the "Select Coding Assistant settings" dialog and click "Finish" button.



Figure 4-7 : Coding Assistant settings

After a little while, the project is created.

e <sup>2</sup> workspace - sample_project/samp	e_project.scfg - e² studio		- 🗆 X
File Edit Navigate Search Proje	ct Renesas Views Run Window Help		
🔦 🞋 🔳 🎋 Debug	🗸 💽 sample_project HardwareDebug 🗸 🌼 🗄 🖛 🔝 🗞 🗸 👼 🖗	a = 16 🕪 💷 😭 🖏 🕹 📲 🥖 🖾 1 🏘	- • • · · · · · · · · · · · · · · · · ·
[h] ▼ 취 ▼ ∜> (> ▼ ⇒ ▼		Quick Access	😭 🔤 C/C++ 📓 Smart Configurator
🏠 Project Explorer 💥 👘 🗖	🏟 sample_project.scfg 🙁		👩 MCU Package 🙁 📃 🗖
E Sample project	Overview information	ݢ 💽	
, <b>C</b>	← General Information	?	
	This editor allows you to modify the settings stored in configuration file (.scfg)		
	Board Allow board and device selection Clocks Allow clock configuration Components Allow software component selection and configuration Pins Allow general pin configuration and pin configuration for selected software component	Application under development Device RTOS + Components + Pins	
	Overview Board Clocks Components Pins Interrupts		▶ Legend
📮 Console 🛛	🚉 🛃 😥 📑 🚍 👻 🗖 🖬 🔝 Configuration Prol	blems 🔀	₽ ~ - □
Smart Configurator Output	0 items	<u>^</u>	
M03000004: File modified: <u>src\</u>	smc_gen\r_config\r_bsp_config.h	Туре	
<	>		
8 9 9			

Figure 4-8 : Project overview

For the details about creating a new e<sup>2</sup> studio project by the Smart Configurator, see "Renesas e<sup>2</sup> studio Smart Configurator User Guide (R20AN0451)".



## 4.2 Confirm BSP version

After creating a project, right-click "r\_bsp" on the "Software component configuration" screen and select "Change version".

		÷. ÷.	Property		Value	
huna filtar taut			<ul> <li>Configurations</li> </ul>		Value	
type inter text			# Startup selec	t	Enable (use BSP startup)	
🗸 🗁 Startup		^	# User stack se	tting	2 stacks	
V 🗁 Generic			# User stack siz	te .	0x1000	
😵 r_bsp		# interrunt star	k size	0x400		
		Change vers	ion		0x400	
				nd output library functions	Enable	
	*	Remove		harget function	Use BSP charget() function	
		Reset to defa	ault	function name	my_sw_charget_function Use BSP charput() function	
		Download a	nd import sample projects	harput function		
Download and			function name	my_sw_charput_function		
		J				

Overview Board Clocks Components Pins Interrupts

Figure 4-9 : Change version

e <sup>2</sup> Change Version				_		×
Version Selection	i i i i i i i i i i i i i i i i i i i					
Select available ver	rsion					
Component name:	r_bsp					
Current version:	5.21					
Available versions:	5.40					~
?	< Back	Next >	Finish		Cancel	
0						

Figure 4-10 : Change Version dialog

Confirm the "Current version" on the "Change Version" dialog.

If you use the BLE FIT module ver 1.01 or later, the r\_bsp needs to be updated to version 5.40 or later. If the r\_bsp update is not made, the following error occurs when the project is built.

../src/smc\_gen/r\_sci\_rx/src/targets/rx23w/r\_sci\_rx23w\_data.c(153):E0520020:Identifier "IR\_SCI8\_ERI8" is undefined

If you select R5F523W8CxLN or R5F523W8DxLN, use version 5.62 or later.



## 4.3 Clock configuration

Configure the clock on the "clock" tab in the Smart Configurator.

If the BLE FIT module is used, set the clock within the following range.

- System clock (ICLK): more than 8MHz
- Peripheral module clock B (PCLKB): more than 8MHz

The Bluetooth LE Protocol Stack is optimized with ICLK: 32MHz, PCLK: 32MHz.

Renesas recommends configuring the frequency of ICLK and PCLKB to 32MHz to maximize the Bluetooth LE performance.



Figure 4-11 : Clock configuration



## 4.4 Adding FIT modules

Click the scfg file in the project and add the FIT modules on "component" tab in the Smart Configurator.

This section takes the case of using the following functions described in "2.2 Software Requirements" as an example.

- Command Line Interface
- Security Data Management
- LED and Switch control

Click "Add component" button <sup>t</sup>, select r\_ble\_rx23w, r\_byteq, r\_cmt\_rx, r\_flash\_rx, r\_gpio\_rx, r\_irq\_rx, r\_lpc\_rx, r\_sci\_rx and click "Finish" button.

📴 New Co	mponent					×
Software (	Component Selection nponent from those available in li	st			ŧ	
Category	All					$\sim$
Function	All					$\sim$
Filter						
Compon	ents	Shor	t Name	Туре	Version	^
🖶 8-Bit	Timer			Code Generator	1.9.0	
🕀 ADC I	Driver	r_s12	2ad_rx	Firmware Integr	4.91	
H ADPC	M compress/de-compress codec	s r_s2_	rx	Firmware Integr	3.04	
🖶 BLE P	rofile Creation			QE Configurator	1.1.0	
🖶 BLE st	tack	r_ble	_rx23w	Firmware Integr	2.30	
🖶 Board	l Support Packages.	r_bsp	0	Firmware Integr	7.00	
🖶 Buses	;			Code Generator	1.10.0	
🖶 Byte-	based circular buffer library.	r_byt	teq	Firmware Integr	2.00	
CAN /	API for the RX200 Series.	r_rsc	an_rx	Firmware Integr	2.32	
Clock	Frequency Accuracy Measureme	nt		Code Generator	1.10.0	
Clock	Synchronous Control Module fo	r r_eep	prom_spi	Firmware Integr	3.02	~
Show of Hide its Descriptio	only latest version ems that have duplicated function n ware component generates two u prise two 8-bit counter channels,	nality nits (unit 0 totaling fo	), unit 1) of an on-cl our channels.	hip 8-bit timer (TMR	) module	^
Download	the latest FIT drivers and middles	ware				$\sim$
<u>Configure</u>	general settings	walt				
?		< Back	Next >	Finish	Cance	I

Figure 4-12 : Software Component Selection



NOTE: When the necessary FIT modules are not found, click [Download the latest FIT drivers and middleware] and download them in accordance with the procedure in [FIT Module Download] dialog.

e²							×
FIT N Sele	Ľ						
	Title RX Family RX Driver Package Ver.1.23	Document No. R01AN4976EJ0123	Rev. Rev.1.23	lssue date 2019-12-16		Selec Desel	ect All
Filte	rred: ☑ Show RX Driver Package only dule Folder Path:						
	C:¥FITModules			Download		Brows	se

Figure 4-13 : FIT Module Download

NOTE: When downloaded FIT modules are not displayed, click out [Hide items that have duplicated functionality].

🔡 New Co	mponent						>
oftware ( Select con	Component Selection aponent from those available in li	st					
Category	All						~
Function All							, ,
Filter							
Compon	antr		Tune		Version		^
Components		T Deivers (s invels er)	Time later	untion To	2.00		
14-DF	t down-counter independent WD	dt pr)	Firmware Integ	ration Te	3.60		
# 8-Bit	Timer	u(_ix)	Code Generator	,	170		
# ADPCM compress/de-compress codec		software library (	Firmware Integ	ration Te	3.04		
# Batte	rv Backup Function Driver. (r. vba	tt rx)	Firmware Integ	ration Te	1.05		
BLE P	BLE Profile Creation		QE Configurator				
Show o	only latest version						
Hide it	ems that have duplicated functio	nality					
Descriptio	n						
Depende This mod generate applicati	ncy : r_bsp version(s) 5.52 lule provides an API for the Indep a reset or NMI signal upon expir on when running. A function is p	endent Watchdog ation. The IWDT ma rovided to periodic	Timer (IWDT). The y be started autor ally refresh the tin	IWDT can be natically at re ner.	e configu eset or b	ured to y the	-
Download Configure	i more software components general settings						
?		< Back	Next >	Finish		Can	cel

Figure 4-14 : Software Component Selection

After a little while, the selected FIT modules are added.





Figure 4-15 : Software component configuration


# 4.5 Configuration options

Configure the options according to the BLE FIT features that you want to use after adding the FIT modules.

# 4.5.1 BSP(r\_bsp)

Set the ID Code to enable the Flash Memory Protection described in "2.10 Device-specific data". Select "Components" tab  $\rightarrow$ "r\_bsp" $\rightarrow$ "ID code 1" and set the option to "0x45FFFFFF".



Figure 4-16 : BSP configuration options



### 4.5.2 Command Line Interface

Set the configuration options for the BLE FIT Module(r\_ble\_rx23w) and the SCI FIT Modules(r\_sci\_rx) for the Command Line Interface. The byte queues/circular buffers (r\_byteq\_rx) need not be set the configuration options.

## (1) BLE(r\_ble\_rx23w)

Select "Components" tab  $\rightarrow$ "r\_ble\_rx23w"  $\rightarrow$ "Enabled/Disabled command line function", set the option to "Enabled". Configure the "SCI CH for command line function" option to the SCI channel number for command line interface. In case of the Target Board (TB) and the RSSK board, set the option to "8".

te 🔁	Property Value	^
type filter text	# CLKOUT_RF output setting. No output	
	# RF_DEEP_SLEEP transition. Enable	
V 🔁 Startup	# MCU Main Clock Frequency (kHz). 4000	
V 🗁 Generic	# Device specific data block on Code Flash (ROM). 16	
💣 r_bsp	# Device specific data block on E2 Data Flash. 7	
V 🔁 Drivers	# MTU Size configured by GATT MTU exchange prc 247	
V 🔁 Interrupt	# Number of remote device bonding information. 7	
r_irq_rx	# Store Security Data in DataFlash Enable	
V 🔁 I/O Ports	# Data Flash Block for Security Data Management. 0	
r_gpio_rx	# Enabled/Disabled command line function. Enable	
Low power consumption	# SCI CH for command line function. 8	
	# Enabled/Disabled board LED and Switch control  Enable	
V 🗁 Memory	# Board Type Customer board	
😭 r_flash_rx	# Log level Error Warning Debu	ig i
Communications	✓ I Resources	
ど r_sci_rx	DE BLE	
V 🔁 Timers		×
<ul> <li>F_cmt_rx</li> <li>Middleware</li> <li>Generic</li> </ul>	Macro definition: BLE_CFG_CMD_LINE_CH SCI CH for command line function.	^
r_ble_rx23w		
🍟 r_byteq		





### (2) SCI(r\_sci\_rx)

Select "Components" tab $\rightarrow$ "r\_sci\_rx" and set the following configuration options.

• Select the "Include software support for channel n" (n is the SCI channel number for command line) and set the option to "Include".

	-		
Property		Value	^
#	Include software support for channel 6	Not	
#	Include software support for channel 7	Not	
#	Include software support for channel 8	Include	
#	Include software support for channel 9	Not	
#	Include software support for channel 10	Not	
#	Include software support for channel 11	Not	
#	Include software support for channel 12	Not	~
SPECIFY CH	ANNELS TO INCLUDE SOFTWARE SUPPORT FOR ing ASYNC mode, adjust BYTEQ_CFG_MAX_CTRL	1=included, 0=not _BLKS in r_byteq_config.h to provide 2 queues per	
channel (s - * = port o	atic mode only). onnector RDKRX63N, RSKRX210, RSKRX11x		

Figure 4-18 : SCI configuration options for Command Line Interface (1)

 Select the "ASYNC mode TX queue buffer size for channel n" (n is the SCI channel number for command line) and set the option to "160".

Prope	erty	Value	^
	# ASYNC mode TX queue buffer size for channel 5	80	
	# ASYNC mode TX queue buffer size for channel 6	80	
	# ASYNC mode TX queue buffer size for channel 7	80	
	# ASYNC mode TX queue buffer size for channel 8	160	
	# ASYNC mode TX queue buffer size for channel 9	80	~
<			>
Macro SPECIF	edefinition: SCI_CFG_CH8_TX_BUFSIZ Y ASYNC MODE TX QUEUE BUFFER SIZES (will not allocat	e if chan not enabled	^
			$\sim$

Figure 4-19 : SCI configuration options for Command Line Interface (2)

Property	Value	
# ASYNC mode RX queue buffer size for channel 11	80	
# ASYNC mode RX queue buffer size for channel 12	80	
# Transmit end interrupt	Enable	
# GROUP12 (Receive error) interrupt priority	3	
# GROUPBL0 (ERI, TEI) interrupt priority	3	
Macro definition: SCI_CFG_TEL_INCLUDED ENABLE TRANSMIT END INTERRUPT (ASYNCHRONOUS) This interrupt only occurs when the last bit of the last byte of ralls the user's callback function specified in R_SCI_Open() and disable an external transceiver to save power. It would then b again. Not including this feature reduces code space used by	data has been sent and the transmitter has become idle. The ir d passes it an SCI_EVT_TEI event. A typical use of this feature is e up to the user's code to re-enable the transceiver before sent the interrupt. Note that this equate is only for including the TE	iterrupt ; to ding El code.

Figure 4-20 : SCI configuration options for Command Line Interface (3)

 Check the SCI channel number, "RXDn/SMISOn" and "TXDn/SMOSIn" (n is the SCI channel number for command line) in "SCI" resources.

Property	Value	^
✓ I Resources		
V 回 SCI		
> 🖽 SCI1		
> 🖽 SCI5		
🗸 🔛 SCI8		
🛰 SCK8 Pin	Unused	
RXD8/SMISO8/SSCL8 Pin	Used	
TXD8/SMOSI8/SSDA8 Pin	Vsed	
CTS8#/RTS8#/SS8# Pin	Unused	×

Figure 4-21 : SCI configuration options for Command Line Interface (4)



#### 4.5.3 Security Data Management/Device-specific data (in data flash)

Set the BLE FIT configuration options for the Security Data Management. It does not need to configure the Data Flash FIT module(r\_flash\_rx) options.

### (1) BLE(r\_ble\_rx23w)

Select "Components" tab  $\rightarrow$ "r\_ble\_rx23w" $\rightarrow$ "Store Security Data in DataFlash" and set the option to "Enabled". And configure the "Data Flash Block for Security Data Management" option to the Data Flash block number that you want to use.

In addition, configure the "Device Specific data block on E2 Data Flash" option to the block number for device-specific data in Data Flash.

Components 🔰 📮 📑 🕇 🔻	Configure	
Image: Components     Image: Components       type filter text       Startup       Seneric       r_cbsp       Drivers       Interrupt       r_irr_rx       V /O Ports       r_gpio_rx       Low power consumption       r_lpc_rx       Memory       r_flash_rx       Communications	Configure         Property       Value         # Sleep Clock Accuracy(SCA) for RF slow clock.       250         # Transmission power maximum value.       max +4dBm         # Default transmit power.       High odBm(BLE_CFG_RF_MAX_TX_POW = 0) / +         # CLKOUT_RF output setting.       No output         # RF_DEEP_SLEEP transition.       Enable         # MCU Main Clock Frequency (kHz).       4000         # Device specific data block on Code Flash (ROM).       16         # Device specific data block on E2 Data Flash.       7         # MIU Size contigured by GATI MIU exchange prc 24/       44/         # Number of remote device bonding information.       7         # Store Security Data in DataFlash       Enable         # Dabled/Disabled command line function.       Enable         # Enabled/Disabled command line function.       8         # Enabled/Disabled command line function.       8         # Enabled/Disabled command line function.       8	
<ul> <li>r_sci_rx</li> <li>Fimers</li> </ul>	Board Type     Customer board      Log level     Error Warning Debug	
<ul> <li>Ber_cmt_rx</li> <li>Middleware</li> <li>Generic</li> <li>Cr_ble_rx23w</li> <li>r_byteq</li> </ul>	Macro definition: BLE_CFG_SECD_DATA_DF_BLOCK Data Flash Block for Security Data Management.	^

Overview Board Clocks Components Pins Interrupts

Figure 4-22 : BLE configuration options for Security Data Management



# 4.5.4 LED and Switch control

Set the BLE(r\_ble\_rx23w) and the IRQ(r\_irq\_rx) configuration options for LED and Switch control. It does not need to configure the GPIO options.

Table 4.1 shows the pins that connects to the LED and the Switch on the Target Board and the RSSK. LED1, LED2, SW1 and SW2 in Table 4.1 are the notations in r\_ble\_board.c. If you use a customer board, the values need to be set in accordance with the board.

Board	LED	Switch
Target Board <sup>*1</sup>	LED1: PC0 LED2: PB0	SW1: IRQ5
RSSK	LED1: P42 LED2: P43	SW1: IRQ1 SW2: IRQ0
Customer board	The user can arbitrarily change the followings. LED1: PC5(default) LED2: PC6(default)	The user can arbitrarily change the followings. SW1: IRQ7(default) SW2: IRQ5 (default)

Table 4.1 : Pins connected to the LED and the Switch.

<sup>\*1</sup>: The LED notations on Target Board are LED0 and LED1 and are corresponding to LED1 and LED2 in Table 4.1.

### (1) BLE(r\_ble\_rx23w)

Select "Components" tab  $\rightarrow$ "r\_ble\_rx23w" $\rightarrow$ " Enabled/Disabled board LED and Switch control support" and set the option to "Enabled" and configure the "Board Type" option to the type of the board that you want to use. If selecting "Customer board, modify the codes described in "4.6.2 LED and Switch control for customer board".



Overview Board Clocks Components Pins Interrupts

Figure 4-23 : BLE configuration options for LED and Switch control



### (2) IRQ(r\_irq\_rx)

Select "Components" tab  $\rightarrow$  "r\_irq\_rx" and set the following configuration options.

• Set the "Filter for IRQn" option (n is the IRQ terminal number for the board switch) to "Enabled" and the "Filter clock divisor for IRQn" option to "Divisor1". Figure 4-24 shows the default values for the default Customer Board. If you use a Customer Board, configure the "Filter for IRQn" option and the "Filter clock divisor for IRQn" in accordance with the board.

Property		Value	^
#	Filter clock divisor for IRQ4	Divisor 1	
#	Filter for IRO5	Enable	
#	Filter clock divisor for IRQ5	Divisor 1	
#	Filter for IRQ6	Disable	
#	Filter clock divisor for IRO6	Divisor 1	
#	Filter for IRQ7	Enable	
#	Filter clock divisor for IRQ7	Divisor 1	
#	Filter for IRQ8	Disable	
#	Filter clock divisor for IRQ8	Divisor 1	~
Macro def To enable Set the val Some devi	inition: IRQ_CFG_FILT_EN_IRQ7 digital noise filtering with the selected IRQ. ue to 1 to enable the filter or 0 to disable it. ces may not support IRQ7. Please confirm with Us	er's Manual: Hardware before adopting this setting.	< >

Figure 4-24 : IRQ configuration options for LED and Switch control (1)

• Check the "IRQn Pin" connected to the board switch in "ICU" resources. Figure 4-25 shows the default for the default Customer Board. Check the boxes according to your board IRQ pins.

Property	Value	^
# Filter clock divisor for IRQ13	Divisor 1	
# Filter for IRQ14	Disable	
# Filter clock divisor for IRQ14	Divisor 1	
# Filter for IRQ15	Disable	
# Filter clock divisor for IRQ15	Divisor 1	
✓ I Resources		
🗸 🗐 ICU		
🛰 IRQ0 Pin	Unused	
🛰 IRQ1 Pin	Unused	
🛰 IRQ4 Pin	Unused	
🛰 IRQ5 Pin	Used	
🛰 IRQ6 Pin	Unused	
🛰 IRQ7 Pin	Used	~
Macro definition: IRQ_CFG_FILT_EN_IRQ7 To enable digital noise filtering with the selected IR0 Set the value to 1 to enable the filter or 0 to disable Some devices may not support IRQ7. Please confirm	). t. with User's Manual: Hardware before adopi	ting this setting.
		~

Figure 4-25 : IRQ configuration options for LED and Switch control (2)



# 4.5.5 Renesas FreeRTOS

## (1) FreeRTOS Kernel

Select "FreeRTOS Kernel" in Smart Configurator "component" tab to configure the FreeRTOS Kernel. If a RX23W device communicates with Bluetooth LE, change the following configuration option from the default setting.

Because the RF interrupt (BLEIRQ) has the maximum priority (14), change the priority to the following.

- configMAX\_PRIORITIES (Maximum number of priorities to the application task) 7 -> 16
- configMAX\_SYSCALL\_INTERRUPT\_PRIORITY (Maximum syscall interrupt priority) 4 -> 15

Components 🕴 🛔 📩 🔻	Configure	
😧 🐨	Property	Value
type filter text	# Use application task tag	Disable
	# Queue set functionality	Enable
Low power consumption	# Counting semaphore functionality	Enable
	# Maximum number of priorities to the application	16
V 🗁 Memory	# Maximum number of priorities to the application	2
🍟 r_flash_rx	# Software timer functionality	📝 Enable
Communications	# The priority of the software timer task	6
🍟 r_sci_rx	# The length of the software timer command queue	5
✓	# The stack depth allocated to the software timer ta	configMINIMAL_STACK_SIZE
皆 r_cmt_rx	# Kernel interrupt priority	1
✓ ➢ Middleware	# Maximum syscall interrupt priority	15
🗸 🗁 Generic	# Tick vector	_CMT0_CMI0
💣 r_ble_qe_utility	# vTaskPrioritySet	V Enable
🚏 r_ble_rx23w	# uxTaskPriorityGet	V Enable
🍟 r_byteq	# vTaskDelete	V Enable
✓ ➢ Application	# vTaskCleanUpResources	Disable
Communications		
Config_BLE_PROFILE	Macro definition configMAX_SYSCALL_INTERRUPT_PRIORITY	
🗸 🗁 RTOS	The maximum interrupt priority from which FreeRTOS API call	s can be made.
🗸 🗁 RTOS Object	Interrupts that use a priority above this will not be effected by	/ anything the kernel is doing.
FreeRTOS_Object		
V 🗁 RTOS Kernel		
FreeRTOS_Kernel		

Figure 4-26 FreeRTOS Kernel Configuration (1)

Configure the following option to change the heap size. It recommends that this option is set to about 10 for Bluetooth LE communication.

• configTOTAL\_HEAP\_SIZE\_N (The configTOTAL\_HEAP\_SIZE\_N)

Components	z 🖹 🕂 井 🔻	Configure		
type filter text		Property  # The frequency of the RTOS tick interrupt # The size of the stack used by the idle task # The configTOTAL HEAP SIZE N # The total amount of RAM available in the FreeRTO # The maximum permissible length of name # Use trace facility """ Macro definition configTOTAL_HEAP_SIZE_N The total amount of RAM available in the PreeRTOS heap(Uni (ize t)) (configTOTAL_HEAP_SIZE_N) The total amount of RAM available in the PreeRTOS heap(Uni (ize t)) (configTOTAL_HEAP_SIZE_N)	Value (TickType_t) 1000 (unsigned short) 140 10 (size_t) (configTOTAL_HEAP_SIZE_N* 1024) 12 II Enable III Enable	ZE(Bytes) in following expression:
<ul> <li>✓ Communications</li> <li>✓ Config_BLE_PROFILE</li> <li>✓ ERTOS</li> <li>✓ RTOS Object</li> <li>✓ RTOS Cobject</li> <li>✓ RTOS Kernel</li> <li>✓ FreeRTOS_Kernel</li> </ul>	×	The maximum permissible length of name     Use trace facility     """     Gitare facility     ""     Gitare facility     Gitare facility	12 Tenable	ZE(Bytes) in following expression





Use the following configurations as default setting.

- configUSE\_MUTEXES(Mutex functionality) 1(Enable)
- configUSE\_TASK\_NOTIFICATIONS (Use task notifications) 1(Enable)

For more information about the other FreeRTOS Kernel configurations, refer to the following documents with the "configXXX" word described in Smart Configurator "Macro definition".

- FreeRTOS customization: https://www.freertos.org/a00110.html
- "RX Family Renesas FreeRTOS (R01AN4307) 3.2 Custom Configuration"

The configuration options are output to src/frtos\_config/FreeRTOSConfig.h .

#### (2) Subtask Declaration

If your application use tasks other than main task, select "FreeRTOS\_Object" in Smart Configurator "component" tab and input the subtask information and the resource that the application uses. This section describes the subtask declaration.

Components $\downarrow^a_Z \boxdot  \Rightarrow$	•	Confi	gure									
ت 👻		He	eap E	stimation	Tasks	Semaphores	Queues Softv	vare Timers Even	t Groups Stream	Buffers	Message Buffers	
type filter text			+/-	Initialize		Task Code	Task Name	Stack Size	Task Handler	Paramete	er Priority	Heap Usage
🗸 🗁 RTOS 🧳			0	manual	~	bas_task	bas_task	256	g_bas_task	conn_h	dl 4	0
V 🗁 RTOS Object			0	manual	~	lss_task	lss_task	256	g_lss_task	conn_h	dl 4	0
FreeRTOS_Object     FreeRTOS Kernel     FreeRTOS Kernel			0									

Figure 4-28 : FreeRTOS Object Configuration

Table **4.2** shows the parameters shown in the FreeRTOS Object Configuration (Figure 4-28).

Parameter	Description	
Initialize	Specify where the subtasks are generated. If "manual" is selected, Object_init_manual() in freertos_object_init.c generates the subtasks. This function needs to be called by application.	
	If "kernel" is selected, Kernel_Object_init() in freertos_object_init.c generates the subtasks. The Renesas FreeRTOS code calls this function.	
Task Code	Subtask function.	
Task Name	Subtask name for debug	
Stack Size	This parameter multiplied by 4 bytes are allocated for the subtask stack size.	
Task Handler	A variable for storing the task handle.	
Parameter	Parameter is passed to the task function in generating the task.	
Priority	Task priority.	



If Bluetooth LE communication is performed, set less than the BLE task priority to the
subtask priority.

After declaring subtasks in FreeRTOS Object Configuration, the files in Table **4.3** are output with code generation.

### Table 4.3 : Codes regarding subtask output in code generation

File	Description
src/frtos_skeleton/"subtask name".c	Subtask template
src/frtos_startup/freertos_object_init.c	Code included in subtask generation function.



# 4.6 Configuration after code generation

Output the codes by click the code generation button 🔞 . Configure the followings after code generation.

## 4.6.1 Changing CMT (if the CMT FIT module version is prior to 4.50)

Because the BLE FIT module use the two CMT channels, add the following after the definition of CMT\_RX\_NUM\_CHANNELS in the r\_cmt\_rx.c in the CMT FIT module. (Figure 4-29)

If the CMT FIT module is v4.50 or later, you do not need to change the source code.

```
#if defined(BSP MCU RX23W)
 #undef CMT RX NUM CHANNELS
 #define CMT RX NUM CHANNELS
                                                                (2)
 #endif /* BSP MCU RX23W */
Macro definitions
                                                                           /* Define the number of CMT channels based on MCU type. */
#if defined(BSP_MCU_RX62_ALL) || defined(BSP_MCU_RX63_ALL) || defined(BSP_MCU_RX21_ALL) || \
    defined(BSP_MCU_RX61_ALL) || defined(BSP_MCU_RX64_ALL) || defined(BSP_MCU_RX113) || \
    defined(BSP_MCU_RX71_ALL) || defined(BSP_MCU_RX231) || defined(BSP_MCU_RX23_ALL) || \

    defined(BSP_MCU_RX71_ALL) || defined(BSP_MCU_RX231) || defined(BSP_MCU_RX23_ALL) || \
defined(BSP_MCU_RX24_ALL) || defined(BSP_MCU_RX65_ALL) || defined(BSP_MCU_RX66_ALL) || \
    defined(BSP_MCU_RX72_ALL)
    #define CMT_RX_NUM_CHANNELS
                                         (4)
#elif defined(BSP_MCU_RX111) || defined(BSP_MCU_RX110) || defined(BSP_MCU_RX130)
    #define CMT_RX_NUM_CHANNELS
                                         (2)
#else
    #error "Error! Number of channels for this MCU is not defined in r_cmt_rx.c"
#endif
#if defined(BSP MCU RX23W)
#undef CMT_RX_NUM_CHANNELS
#define CMT_RX_NUM_CHANNELS
                                         (2)
#endif /* BSP MCU RX23W
```





Г

# 4.6.2 LED and Switch control for customer board

In case of using a customer board, modify the followings in app\_lib/board/r\_ble\_board.c.

#### (1) Macros of LED and Switch

Modify the below macros which define the IRQ and the GPIO pin number in accordance with the customer board.

BLE\_BOARD\_SW1\_IRQ BLE\_BOARD\_SW2\_IRQ BLE\_BOARD\_LED1\_PIN BLE\_BOARD\_LED2\_PIN

f (PLE CEC POARD TYPE 1) /* for PY224 Tanget Poard(TP) */			
(DLE_CFG_BOARD_TYPE == 1) /* TOP RAZSW Tanget Board(TB) */			
lefine BLE_BOARD_SW1_IRQ (IRQ_NUM_5)			
<pre>lefine BLE_BOARD_SW2_IRQ (IRQ_NUM_5)</pre>			
<pre>lefine BLE_BOARD_LED1_PIN (GPI0_PORT_C_PIN_0)</pre>			
<pre>lefine BLE_BOARD_LED2_PIN (GPIO_PORT_B_PIN_0)</pre>			
<pre>elif (BLE_CFG_BOARD_TYPE == 2) /* for RX23W RSSK board */</pre>			
<pre>lefine BLE_BOARD_SW1_IRQ (IRQ_NUM_1)</pre>			
<pre>lefine BLE_BOARD_SW2_IRQ (IRQ_NUM_0)</pre>			
#define BLE BOARD LED1 PIN (GPIO PORT 4 PIN 2)			
#define BLE BOARD LED2 PIN (GPIO PORT 4 PIN 3)			
else /* BLE_CFG_BOARD_TYPE */			
lefine BLE_BOARD_SW1_IRQ (IRQ_NUM_7)			
lefine BLE_BOARD_SW2_IRQ (IRQ_NUM_5) If you use a customer board,			
lefine BLE_BOARD_LED1_PIN (GPIO_PORT_C_PIN_5) these values need to be set in	I		
#define BLE_BOARD_LED2_PIN (GPIO_PORT_C_PIN_6)			
endif /* BLE CFG BOARD TYPE */			

Figure 4-30 : LED, SW macros

If SW1 connects to IRQ7, SW2 connects to IRQ5, LED1 pin connects to PortC pin5 and LED2 pin connects to PortC pin6, make modifications shown in Figure 4-30.



# (2) Configure the registers in irq\_pin\_set()

Modify the register settings in irq\_pin\_set() for the IRQ pins connected to the SW on the customer board.



Figure 4-31 : The changes in irq\_pin\_set()



# 4.6.3 Add application code

When a project is created, the application code ("{Project Name}.c") is generated. But it is a blank template. Therefore, modify main() to call app\_main() in app\_main.c generated by the QE for BLE. If you use Renesas FreeRTOS, modify main\_task() to call app\_main(). Refer to "6.2.1 Main Task (2) call app\_main()". Here is a sample main().

Figure 4-32 : Sample main() calling app\_main()

### (1) QE for BLE

Select and configure profiles, services, characteristics in the QE for BLE, output the base code for the application and profile development. For the QE for BLE details, see "Bluetooth Low Energy Profile Developer's Guide(R01AN6459)".



## 4.6.4 **Profile Common code generation**

The BLE FIT v2.50 or later doesn't include Profile Common. Profile Common is provided by QE for BLE (V1.6.0). If you use the BLE FIT and QE for BLE in the following combination, build errors due to absence of Profile Common code occurs.

- BLE FIT v2.50 or later
- before QE for BLE V1.6.0

In this situation, it is necessary to add Profile Common code to your project by one of the following processes.

- Code generation after update QE for BLE to V1.6.0. Update QE for BLE to V1.6.0 according to "Installing This Product" in QE for BLE[RA, RE, RX] V1.6.0 Release Note at <u>QE for BLE: Development Assistance Tool for Bluetooth Low Energy</u>. After updating, when code generation is executed with QE for BLE, Profile Common code is added to your project.
- Copy Profile Common code from trash. If you update the BLE FIT from before v2.50 to v2.50 or later, the previous BLE FIT moves to the trash directory. Copy the previous version Profile Common code (profile\_cmn, discovery) and paste the copy in ge gen/ble directory.



Figure 4-33: Copy Profile Common code



# 4.7 Linker configurations

# 4.7.1 Bluetooth LE Protocol Stack program section separation

Configure the section for the BLE FIT.

Table 4.4 shows the section name of the Bluetooth LE Protocol Stack.

No.	Name	Default Section Name	Bluetooth LE Protocol Stack Section Name	Description
1	Program area (ROM)	P	BLE_P	Stores machine code
		С	BLE_C	Stores const type data
2	Constant area (ROM)	C_2	BLE_C_2	
		C_1	BLE_C_1	
		D	BLE_D	Stores data with initial values in ROM
3	Initialized data area (ROM)	D_2	BLE_D_2	
		D_1	BLE_D_1	
		R	BLE_R	Stores data with initial values in RAM
4	Initialized data area (RAM)	R_2	BLE_R_2	
		R_1	BLE_R_1	
		В	BLE_B	Stores data without initial values
5	Uninitialized data area (RAM)	B_2	BLE_B_2	
		B_1	BLE_B_1	
		W	BLE_W	Stores branch tables for switch
6	Switch statement branch table area (ROM)	W_2	BLE_W_2	
		W_1	BLE_W_1	
7	Literal area (ROM)	L	BLE_L	Stores string literals and initializers used for dynamic initialization of aggregates

Table 4.4 : 5	Section Name
---------------	--------------

Set the section name shown in Table 4.4 to the Bluetooth LE Application project in the following procedure.



### 4.7.1.1 Add the section name

- (1) Click [Project]  $\rightarrow$  [Properties]  $\rightarrow$ [C/C++ build]  $\rightarrow$  [Settings].
- (2) Select "Linker"  $\rightarrow$  "Section" in the "Tool Settings" tab and click the "..." button.
- (3) Input the following in the "Section Viewer".

RAM	BLE_B* BLE_R*
ROM	BLE_C* BLE_D* BLE_W* BLE_L BLE P



Figure 4-34 : Section Configuration



## 4.7.1.2 Rom to RAM mapped section

- (1) Select "Settings"  $\rightarrow$  "Tool Settings"  $\rightarrow$  "Linker"  $\rightarrow$  "Section"  $\rightarrow$  "Symbol file".
- (2) Input the following to the "ROM to RAM mapped section". (Figure 4-35)



Figure 4-35 : Rom to RAM mapped section configuration

#### NOTE:

Because the R\_BLE\_Open() initialize the BLE section, it is not necessary to change the DTBL/BTBL table.



# 4.7.2 Bluetooth LE Protocol Stack Library Configuration

The Bluetooth LE Protocol Stack is provided as static library. Three types of libraries shown in Table 4.5 are included in the lib directory of the BLE FIT module.

Table 4.5 : Bluetooth LE	<b>Protocol Stack libraries</b>
--------------------------	---------------------------------

Type of the Bluetooth LE Protocol Stack	Library
All features	lib_ble_ps_ccrx_a.lib
Balance	lib_ble_ps_ccrx_b.lib
Compact	lib_ble_ps_ccrx_c.lib

Configure the followings to make the Bluetooth LE Application project available to the Bluetooth LE Protocol Stack library.

#### 1) Linker configuration

Click [Project]  $\rightarrow$  [Properties] on e<sup>2</sup>studio.

Select "C/C++ Build"  $\rightarrow$  "Settings"  $\rightarrow$  "Tool Settings"  $\rightarrow$  "Linker"  $\rightarrow$  "Input".

Confirm that the following library is added to the "Relocatable files, object files and library files" (Figure 4-36). If not, add the library.

"\${workspace\_loc:/\${ProjName}/src/smc\_gen/r\_ble\_rx23w/lib/lib\_ble\_ps\_ccrx.lib}"

e <sup>2</sup> Properties for sample_profile	server		_		×
type filter text  Resource Builders	Settings	Relocatable files, object files and library files		(> ▼ ↔	• •
<ul> <li>C/C++ Build Build Variables Environment Logging Settings Tool Chain Editor</li> <li>C/C++ General Project References Renesas QE Run/Debug Settings SVN Info</li> <li>Task Repository</li> </ul>	<ul> <li>&gt; So Compiler</li> <li>&gt; So Assembler</li> <li>&gt; Compiler</li> <li>&gt; So Assembler</li> <li>&gt; Compiler</li> <li>&gt; Advanced</li> <li>&gt; Advanced</li> <li>&gt; Advanced</li> <li>&gt; List</li> <li>&gt; Optimization</li> <li>&gt; Section</li> <li>&gt; Symbol file</li> <li>&gt; Advanced</li> <li>&gt; Subcommand file</li> <li>&gt; Miscellaneous</li> <li>&gt; User</li> <li>&gt; Library Generator</li> </ul>	*\${workspace_loc:/\${ProjName}/src/smc_gen/r_ble_23w/lib/li	b_ble_ps_cd	:rx.lib)"	~
?		Apply an	d Close	Cancel	

Figure 4-36 : Linker configuration



#### 2) Register the bat file

Register the bat file to switch the referred library in conjunction with the configuration option of the BLE FIT module.

Click [Project]  $\rightarrow$  [Properties] on e<sup>2</sup> studio.

Select "C/C++ Build"  $\rightarrow$  "Settings"  $\rightarrow$  "Build Steps".

Input the following to the "Command(s)" in the "Pre-build steps". (See Figure 4-37)

{ProjDirPath}/src/smc\_gen/r\_ble\_rx23w/lib/ble\_fit\_lib\_selector.bat

Properties for sample_project	— 🗆 X
type filter text	Settings $\diamondsuit \checkmark \checkmark \diamondsuit$
<ul> <li>&gt; Resource Builders</li> <li>&gt; C/C++ Build Build Variables Environment Logging Settings</li> <li>Stack Analysis Tool Chain Editor</li> <li>&gt; C/C++ General Project Natures Project References Refactoring History</li> </ul>	Configuration: HardwareDebug [Active]
	Iool Settings       Ioolchain       Device       Build Steps       Build Artifact       Binary Parsers       Binary Parsers       Errc         Pre-build steps       Command(s):       \$(ProjDirPath)/src/smc_gen/r_ble_rx23w/lib/ble_fit_lib_selector.bat       V         Description:       V
Renesas QE Run/Debug Settings	Post-build steps       Command(s):       V       Description:       V
?	Apply and Close Cancel

Figure 4-37 : Build Steps configuration



# 4.8 Debug configurations

Select "Run"  $\rightarrow$  "Debug Configurations..."  $\rightarrow$  "Renesas GDB Hardware Debugging"  $\rightarrow$  the application project and configure the followings. Build the project after setting the debug configurations. If the build is successfully finished, write the firmware to the board.

# 4.8.1 Flash ID Code

Click [RUN] $\rightarrow$ [Debug Configuration] on the e<sup>2</sup>studio.

If writing a firmware failed, confirm the "ID Code".

e <sup>2</sup> Debug Configurations			×
Create, manage, and run configurations [Main]: Program does not exist			Ť.
Image: Second Secon	Name: sample_project HardwareDebug  Main Stantup Source Debug hardware: E1 (RX)  GDB Settings Connection Settings Debug Targ  GDB Settings Connection Settings Debug Targ  GDB Settings Connection Settings Debug Targ  Change startup bank Startup bank  Communication Mode Mode Execute The User Program After Ending	ce Common et Device: R5F523W8 ool Settings Single Chip Single-chip mode No Bank 0 Debug Mode No	
Renesas Simulator Debugging (RX, RL78) Filter matched 14 of 16 items	ID Code	45FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	pply Close

Figure 4-38 : ID Code protection configuration



# 4.8.2 Power

Configure "Power Target From The Emulator (MAX 200mA)" to "No".

e Debug Configurations			×
Create, manage, and run configurations			-
😵 [Main]: Program does not exist			
	Name: sample_project HardwareDebug		
type filter text	🐴 Main 🕸 Debugger 🌔 Startup 🔲 Common	ty Source	
C/C++ Application     C/C++ Remote Application     EASE Script	Debug hardware: E1 (RX) V Target De	evice: R5F523W8	
GDB Finulator Debugging	GDB Settings Connection Settings Debug Tool Se	ettings	
Java Applet	Jlag Clock Frequency[MHz]	16.5	<u> </u>
Java Application	Fine Baud Rate[Mbps]	2.00	
🔁 Launch Group	Hot Plug	NO	
Launch Group (Deprecated)	Power Target From The Emulator (MAX 200m	A) No	
Remote Java Application	Supply Voltage	3.3V	
✓ C <sup>™</sup> Renesas GDB Hardware Debugging	CPII Operating Mode	5.54	
ble_demo_tbrx23w_profile_client	Register Setting	Single Chip	
sample_project HardwareDebug	Mode pin	Single-chip mode	
Renesas Simulator Debugging (RX, RL78)	Change startup bank	No	· · · · · · · · · · · · · · · · · · ·
	Startup bank	Bank 0	v <b>v</b>
Filter matched 14 of 16 items		Revert	Apply
?		Debug	Close

Figure 4-39 : Power supply from the emulator



# 4.9 IAR development environments

This section describes the project creation on the IAR Embedded Workbench for Renesas RX. If a BLE project includes Renesas FreeRTOS, it does not support IAR development environments.

### 4.9.1 Create a project

Create a new CCRX project on e<sup>2</sup> studio according to Section 4.1 to 4.7.

### 4.9.2 Add iodefines.h for IAR

Generate the iodefines.h by the way (1) or (2) and add it in the CCRX project. The iodefines.h needs to be added to the CCRX project before "4.9.3 Project conversion".

- (1) Generate from the RX Smart Configurator
  - Install the RX Smart Configurator from <u>https://www.renesas.com/software-tool/smart-configurator</u>.
  - Start the RX Smart Configurator.
  - Select "File >> New...", set the following items and click "Finish".

"Platform": RX23W device "Toolchain": IAR EWRX Toolchain "File name": [Project name]

💰 New Sr	mart Configuration Fi	le				×
Smart Cor	nfiguration Setting	qs				
Select pla	tform and toolchain s	settings for the ne	w configurat	ion file		
Category:	RX					~
Platform:			Toolchain:			
<ul> <li>&gt; RX23I</li> <li>&gt; RX231</li> <li>~ RX231</li> <li>~ RX</li> <li>~ RX</li> <li>&gt; RX</li> </ul>	E-A W K23W - 56pin R5F523W7AxNG R5F523W7BxNG R5F523W8AxNG R5F523W8BxNG K23W - 85pin	< >	Renesz	as RXC Toolchain /RX Toolchain		
ROM size:	512 Kbytes, RAM siz	e: 64 Kbytes, Pin c	ount: 56			
File name:	iar_ble_project					
Location:	C:¥iar_project				В	owse
		< Back	Next >	Finish	Ca	ncel

Figure 4-40 : RX Smart Configurator New Project settings



• Click "Add component" on the "Component" tab, select "r\_bsp" on the "New component" window and click "Finish".



Figure 4-41 : Add component (1)

🔥 New Compo	nent		_		×
Software Component	ponent Selection ent from those availa	ble in list			8 8 8 8
Type All Function All					~
Filter					
Components	^ N	Туре	Version		^
r bsp	v	FIT	5.52		<b>1</b> 11
🖶 r_byteq <		FIT	1.80		>
Show only la Description	atest version				
This software ( (TMR) module	component generate that comprise two 8	s two units (unit 0, bit counter channe	unit 1) of an on-chip els, totaling four char	8-bit tim inels.	er 🔨
Developed even	0				~
Configure gene	e sottware componei eral settings				
<u>compare gene</u>	and a second				
?	< Back	Next >	Finish	Canc	el

Figure 4-42 : Add component (2)

If the "r\_bsp" is not shown, start CS+ and run Smart Configurator to install the "r\_bsp". Refer to "RX Smart Configurator User's Guide: CS+ (R20AN0470)"



• Click "Add component" on the "Component" tab, select "r\_bsp" on the "New component" window and click "Finish".

🔇 Smart Configurator		_	
File Window Help			
			🖻 🛛 🜋
🌼 *iar_ble_project_en.scfg 🛛			
Software component	configuration	"Generate Code" button	6
Co ↓ª <sub>Z</sub> 🖃 🕀 📫 🕇 ▼	Configure		
ت ئ	Property	Value	^
type filter text	✓		-
🗙 🗁 Startup	# Startup select	Enable (use BSP startup)	
A Generic	# User stack setting	2 stacks	
	# User stack size	0x1000	
er_osp	# Interrupt stack size	0x400	
	# Heap size	0x400	$\sim$
	<		>
			0
Overview Board Clocks Comp	oonents Pins Interrupts		

Figure 4-43 : Generate Code

- Copy the "iccrx" directory included in \src\smc\_gen\r\_bsp\mcu\rx23w\register\_access which is in the RX Smart Configurator project location directory (refer Figure 4-40) to \src\smc\_gen\r\_bsp\mcu\rx23w\register\_access which is generated according to 4.9.1.
- (2) Copy from the FITDemos
   Copy the "iccrx" directory included in
   <<Demo Project Name>>\src\smc\_gen\r\_bsp\mcu\rx23w\register\_access which is in the Table 8.1
   demo project to \src\smc\_gen\r\_bsp\mcu\rx23w\register\_access which is generated according to 4.9.1.



#### 4.9.3 Project conversion

Convert a CCRX project to an IAR project according to the following procedure.

- (1) Start the IAR Embedded Workbench for Renesas RX.
- (2) Click "Tools" >> "Convert To IAR for RX...".
- (3) Input the project created by 4.9.1 into "Root directory of source project".
- (4) Check the "Enable" box in "Project file conversion".
- (5) Select "Renesas e2studio for RX" in "Project type".
- (6) Uncheck the "Enable" box in "Source code substitution".
- (7) Click the "Execute" button.

lAR Project Converter - 4.0.2			×
Root directory of source project C:¥projects¥FITDemos¥ble_dem	( <mark>3)</mark> no_tbrx23w_profile_server		
Project file conversion (4) Enable Project type Renesas e2studio for RX		(5)	
File extension(s) *.cproject			
Enable (6) T Substitution rules (the rules will N From	ag changed code with co I be applied as ordered in To	mment: the	Add
<ul> <li>N From</li> <li>1 [¥t]*#pragma[¥t]+inte</li> <li>2 #pragma +segment *?</li> </ul>	To #pragma vector=\$1¥n #pragma segment="\$1"	····	Add Copy
3 #pragma¥s+pack(?!(opt 4sectop 5secend	<pre>#pragma pack(1)section_begin section_end</pre>	•	Edit
6secsize 7 setpsw_i	section_size enable_interrupt	•	Remove
8 clrpsw_i <	disable_interrupt	>	Export
Perform substitutions on file(s) *.c;*.cpp;*.h;*.hpp;*.asm;*.s	with extension(s)	~	
	[	(7) Execute	Close

Figure 4-44 : IAR Project Converter settings screen (1)

(8) Select the "Operate on a copy of the source code in the following directory" button in the "Choose destination directory" window.



#### (9) Input the directory that the converted IAR project are located in and click "OK".



Figure 4-45 : IAR Project Converter settings screen (2)

#### (10) Click "OK" in the "Report" window.

Report -	$\times$
This file: C:¥iar_project¥ble_demo_tbrx23w_profile_server¥ConvertToIARReport.txt	 ^
Project file conversion(s)	
C:¥iar_project¥ble_demo_tbrx23w_profile_server¥.cproject	
C:¥iar_project¥ble_demo_tbrx23w_profile_server¥.cproject => Renesas setup files were excluded. Project converted to => C:¥iar_project¥ble_demo_tbrx23w_profile_server¥ble_demo_tbrx23w_profile_server.ewp Workspace created => C:¥iar_project¥ble_demo_tbrx23w_profile_server¥ble_demo_tbrx23w_profile_server.eww	
Source code substitutions	
	(10)
	OK

Figure 4-46 : IAR Project Converter settings screen (3)



## 4.9.4 Project options configuration

Click "Project >> Add Existing Project..." in the IAR Embedded Workbench for Renesas RX and open the project file (.ewp) converted in 4.9.3. Click "Project >> Option" to change the following options.

### (1) Stack/Heap

Click "General Options >> Stack/Heap". Set "User mode stack size" and "Heap size" and "Supervisor mode stack size" same as the sizes in the CCRX project r\_bsp configuration options. Table **4.6** and Figure **4-47** show the stack/heap relationship between CCRX Project and IAR Project.

CCRX Project	IAR Project
User stack size	User mode stack size
interrupt stack size	Supervisor mode stack size
Heap size	Heap size

#### CCRX Project (r\_bsp configuration options)







#### (2) Target

Click "General Options >> Target" and select a "RX23W Group" device in the "Device". Click the "32 bits" button as the "Size of type 'double' "in the "Floating-point".

Options for node "ble_demo_t	brx23w_profile_server"	Generic	×
		RX110 Group	>
Category:		RX111 Group	>
General Options		RX113 Group	>
Static Analysis		RX130 Group	>
Runtime Checking	Liberry Octions 2 Shade/Ulars	RX13T Group	>
C/C++ Compiler	Target Output Libra	RX210 Group	> A-C: 1998
Assembler	output Libra	RX21A Group	> puons r
Output Converter	RsFs23W8	RX220 Group	>
Build Actions		RX230 Group	>
Linker		RX231 Group	>
Debugger		RX23E-A Group	>
E1/E20		RX23T Group	>
E2 E2 Lite / E7-CLIBE2	32 bits     64 bits	RX23W Group	> R5F523W7
J-Link	Subnormal numbers	RX24T Group	> R5F523W8
Simulator	○ Handle with exceptions	RX24U Group	>
	Treat as zero	RX610 Group	>
	Position-independence	RX621 Group	>
	Code and read-only data	RX62G Group	>
	Read-write data	RX62N Group	>
		RX62T Group	>
		BYC20 Crown	

Figure 4-48 : Target setting

#### (3) Library Configuration

Click "General Options >> Library Configuration". Select "Normal DLIB" on "Library".

Options for node "sample_p	rofile_server"			×
Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Output Converter Custom Build Build Actions Linker Debugger E1/E20 E2 E2 Lite J-Link	Library Options 2 Target Output Library: Normal DLIB ~	Stack/Heap ut Library C Description: Use the norr runtime libra file descripte scant, and no	MISRA-C:2004 Configuration mal configuration o ary. No locale interfor or support, no mult o hex floats in strto	MISRA-C: 1998 Library Options 1 f the C/C++ ace, C locale, no ibytes in printf and d.
Simulator	Configuration file: STOOLKIT_DIRS¥LIB¥ Enable thread supp Suppress the core r	dlocfiln.h <b>port in library</b> runtime model attr	ribute	
			ОК	Cancel

Figure 4-49 : Library Configuration



#### (4) Preprocessor

Click "C/C++ Compiler >> Preprocessor" and confirm that the same include paths as the CCRX Project are input in the "Additional include directories".

Options for node "ble_demo_	tbrx23w_profile_se	ver"				×
Category:					Fac	tory Settings
General Options	Multi-file Compile	ation				
Static Analysis	Discard Un	used Publics				
Runtime Checking	MISRA-C:	1998	Encodings		Extra Op	otions
Assembler	Language 1	Language 2	Code	Optimi	zations	Output
Output Converter	List	Preprocessor	Diagr	ostics	MISRA	A-C:2004
Output Converter Custom Build Build Actions Linker Debugger E1 / E20 E2 E2 Lite / E2-CUBE2 J-Link Simulator	dditional incl \$PROJ_DIR\$/5 \$PROJ_DIR\$/5 \$PROJ_DIR\$/5 \$PROJ_DIR\$/5 \$PROJ_DIR\$/5 Preinclude files Defined symbol	dard include direc ude directories: (r rc/smc_gen/Coni rc/smc_gen/r_ble rc/smc_gen/r_ble rc/smc_gen/r_ble rc/smc_gen/r_ble lis: (one per line)	tories one per line) ig_BLE_PROFIL _qe_utility _rx23w _rx23w/src	E eprocessor ] Preserve c ] Generate	r output to comments #line direct	• ··· • ··· file
			(	)K	Cancel	

Figure 4-50 : Additional include path setting

#### (5) Optimizations

Click "C/C++ Compiler >> Optimizations". Select optimization level. If you don't debug the project, the ROM size can be reduced by configuring the optimization level to "High" and "Size".

Options for node "ble_demo_	tbrx23w_profile_ser	ver"					×
Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger E1 / E20 E2 E2 Lite / EZ-CUBE2 J-Link Simulator	Multi-file Compile Discard Unu MISRA-C: List Language 1	ation used Publics 1998 Preproce Languag	essor ge 2	ncodings Diagn Code abled transfi Loop unrol Function in Code motic Type-basec Cross call Static clusti	ostics Optimi: ubexpressi ling lining a dias anal ering	Fa Extra Q MISE zations on elimin. ysis	ctory Settings rptions VA-C:2004 Output ation
				01	К	Cancel	

Figure 4-51 : Optimization setting



# (6) Output Converter

Click "Output Converter". Check the "Generate additional output" box and select the "Motorola S-records" in the "Output format".

Options for node "ble_dem	p_tbrx23w_profile_server"	×
Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger E1 / E20 E2 E2 Lite / EZ-CUBE2 J-Link Simulator	Output  Generate additional output  Output format:  Output file  Override default  ble_demo_tbrx23w_profile_server.srec	Factory Settings
	OK Ca	ncel

Figure 4-52 : Output Converter setting

#### (7) Pre-build command line

Click "Build Actions". Enter the file path to the ble\_fit\_lib\_selector.bat file on "Pre-build command line".

#### Example:

The bat file path is \$PROJ\_DIR\$\src\smc\_gen\r\_ble\_rx23w\lib\ble\_fit\_lib\_selector.bat.

Options for node "ble_demo_tbrx2	3w_profile_server"	×
Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions P Linker Debugger E1 / E20 E2 E2 Lite / EZ-CUBE2 J-Link Simulator	ild Actions Configuration re-build command line: \$PROJ_DIR\$¥src¥smc_gen¥r_ble_rx23w¥lib¥ble_fit_lib_selector.bat rost-build command line:	
	OK Cancel	

Figure 4-53 : Pre-build command line setting



#### (8) Linker configuration file

Click "Linker >> Config". Check "Override default" and enter the file path to Inkr5f523w8.icf or Inkr5f523w7.icf.

#### Example:

The Inkr5f523w8.icf file path is \$TOOLKIT\_DIR\$\CONFIG\Inkr5f523w8.icf.

Options for node "ble_demo_tbrx23w_profile_server" X
Options for node "ble_demo_tbrx23w_profile_server"       X         Category:       Factory Settings         General Options       Static Analysis         Runtime Checking       C/C++ Compiler         Output Converter       Config Library Input Optimizations Advanced Output List         Unker       Override default         Debugger       E1/E20         E2       E2 Litle /EZ-CUBE2         J.Link       Simulator
DK Cancel

Figure 4-54 : Linker configuration file

#### (9) Additional libraries

Click "Linker >> Library". Enter the library file path on "Additional libraries".

#### Example:

The library file path is \$PROJ\_DIR\$/src/smc\_gen/r\_ble\_rx23w/lib/lib\_ble\_ps\_ccrx.lib.

Options for node "ble_demo_	tbrx23w_prot	file_server					×
Options for node "ble_demo_ Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger E1 / E20 E2 E2 Lite / EZ-CUBE2 J-Link Simulator	#define Config Autor SPROJ Overr © Er N	Diag Library matic runt clude C-S ] Bufferer al librarier ide defaul htty symbol symbo	nostics Input ime library PY debuge d write s: (one per imc_gen/r, it program ol	Checksum Optimizations / selection ging support line) ble_rx23w/lib/lib, iar_program_star	Encodings Advanced ble_ps_ccrx.lil	Extra Output	y Settings Options List
L					OK (	Cancel	

Figure 4-55 : Additional libraries setting



#### (10) Debugger

Click "Debugger >> Setup". Select the emulator on "Driver".

#### Example:

If you use the E2 Lite as the emulator, select "E2 Lite/EX-CUBE2".

Figure 4-56 : Debugger setting



## 4.9.5 Exclude some directories from the build target

Remove the following directories included in the build target due to project conversion.

(1) The lib directory included in the BLE FIT module

Exclude the BLE FIT module lib directory from the build target on GUI. Right-click on the "src/smc\_gen/r\_ble\_rx23w/lib" directory and select "Options...". On the Option window, check the "Exclude from build" box.

HardwareDebug		
Files	Options for node "lib"	$\times$
Die_demo_torxzow_prome_server - Hardwar •		
	Category:	
	State Analysis	
III ⊢⊞ ∎r ble ge utility		
-⊟ ∎r ble rx23w	C-STAT Static Analysis	
	Assembler	
lib_ble_ps_ccrx.lib	Output Converter	
- D lib_ble_ps_ccrx_a.lib	Custom Build Select C-STAT Checks	
l lib_ble_ps_ccrx_b.lib		
l lib_ble_ps_ccrx_c.lib	Import Settings	
l		
l	Export Settings	
	Enable parallel analysis: 2 processes	
Sic	Enable module timeout: 600 seconds	
	Enable false-positives analysis	
	Limit messages per check and file: 100 messages	
ble_demo_tbrx23w_profile_server		
Debug Log	There is no C-STAT license available for this product version.	
Log		
	OK Cancel	

Figure 4-57 : Lib directory option

#### (2) Trash directory

If the build target includes the "trash" directory, right-click the directory and select "Options…". On the Option window, check the "Exclude from build" box.

Workspace 👻 🗜 🗙			
HardwareDebug ~	Options for node "trash"		X
Files 🌣 🔹	Exclude from build		
ble_demo_tb	Category: Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build	C-STAT Static Analysis Select C-STAT Checks Import Settings	
ble_demo_tbrx23w_profile_server		Enable parallel analyzis:	2 processes
Debug Log		Chable parallel analysis:	2 processes
		Enable module timeout:	600 seconds
Log		Enable false-positives analysis	
		✓ Limit messages per check and file:	100 messages
		There is no C-STAT license available for this	product version.
<			
Build Debug Log			OK Cancel

Figure 4-58 : Trash directory option



### RX23W Group

# 4.9.6 Build the project and download the firmware

Click "Project >> Rebuild All" to build the project.

After completion of the build, select "[emulator name] >> Hardware setup ...".

The following sample use the "E2/E2 Lite" as emulator.

ble_demo_tbrx23w_profil	e_server - IAR Embedded Workbench IDE -	RX 4.14.1
File Edit View Project	E2/E2 Lite Tools Window Help	
i 🗅 🗅 🗳 🖆 🗎 🛛 🗶	👫 Hardware Setup	- < Q > ⇆ म्म < 😲
Workspace	<b>℃</b> Operating Frequency	▲ 廿 X
HardwareDebug	Leave Target Running	~
Files	Trace Settings	• ^
🗆 🌒 ble_demo_tbrx23	🔎 Trace	
⊢⊟ ■ src	Function Trace	
Config_BLE	Function Profiler	•
general	Power Log Setup	•
r_ble_rx23w	Power Log	
lib	Timeline	
-⊞ ∎iei		•
r_ble_rx2	Data Sample Setup	
IIII ⊑ ≣ readme.b		
ble demo tbrx23w profile serv	Sampled Graphs	
Debug Los	Performance Analysis Setup	
	Performance Analysis	
Log	Start/Stop Function Settings	
	Download Firmware	
	🖭 Breakpoint Usage	
	Emulator Information	

Figure 4-59 : Hardware setup (1)

Configure the hardware settings on the "Hardware Setup" window. If there is no need to change from the default settings, just click "OK".

Hardware Setup, device: R5F523W8 (E2E2L¥RX23WGR¥R5F523W8.MCU)	×
Hardware Setup, device: RSF523W8 (E2E2L¥RX23WGR¥RSF523W8.MCU) MCU External Memory MCU Single-chip mode Single-chip mode Change statup bank No Change statup bank No Statup bank Bank 0 External memory areas: A Byte order BUS Wi Edit	Communication  JTAG clock:  FINE baud rate:  System  Debug the program re-writing the PROGRAM ROM Debug the program re-writing the DATA FLASH Emulator mode: Trace  Download  Erase flash ROM before download Erase data flash ROM before download  Power target from the emulator (MAX 200mA)  1.8V  3.3V
Allow clock source change when writing internal flash	© 3.3 V 5.0 V
Work RAM start address (0x500 bytes used): 0x1000 Factory Settings	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Figure 4-60 : Hardware setup (2)

Click "Project >> Download" and download the firmware.



# 5. Application Guide

For more information about how to create the Bluetooth LE Application, refer to "RX23W Group Bluetooth Low Energy Application Developer's Guide (R01AN5504)".



# 6. Renesas FreeRTOS BLE task

It assumes that Renesas FreeRTOS Bluetooth LE Application has the task configuration as Figure **6-1**. The BLE task consists of the Bluetooth LE Core Stack features, the Bluetooth LE communication control and the application processing callbacks. The BLE GATT App task consists of the GATT service applications.



Figure 6-1 : Task configuration on Renesas FreeRTOS

# 6.1 Call available API

Call available APIs vary between the BLE task and the BLE GATT App task.

- Call available APIs from the BLE task
  - GAP API (R\_BLE\_GAP\_XXX)
  - GATT Server API (R\_BLE\_GATTS\_XXX)
  - GATT Client API (R\_BLE\_GATTC\_XXX)
  - Vendor Specific API (R\_BLE\_VS\_XXX)
  - L2CAP API (R\_BLE\_L2CAP\_XXX)
  - Abstraction API (R\_BLE\_ABS\_XXX)
  - Discovery API (R\_BLE\_DISC\_XXX)
  - R\_BLE\_[GATT service abbreviation + S(server selected) or C(client selected)]\_YYY
- Call available APIs from the BLE GATT App task
  - R\_BLE\_[GATT service abbreviation + S(server selected) or C(client selected)]\_YYY
- Call available API from either task
  - R\_BLE\_CLI\_Printf

All tasks can call BLE\_LOG\_XXX, APIs using the FIT module other than BLE (R\_BLE\_LPC\_XXX, R\_BLE\_TIMER\_XXX, R\_BLE\_SECD\_XXX, R\_BLE\_CLI\_XXX(except R\_BLE\_CLI\_Printf), R\_BLE\_CMD\_XXX, R\_BLE\_BOARD\_XXX).

When the BLE GATT App task calls the GATT service API, the GATT communication is processed with the Core Stack in the BLE task. Figure 6-2 shows Bluetooth LE communication that two BLE GATT App tasks control the GATT services and send notification of the GATT service characteristic on the multi-task environment including the BLE task and the BLE GATT App task.



Figure 6-2 : BLE task and BLE GATT App task


## 6.2 BLE task generation

The BLE task is generated as main task or a task other than main task (hereafter referred to as the subtask).

#### 6.2.1 Generation as main task

If the BLE task is generated as main task, change the following code.

#### (1) Change the parameters of main task generation

The main task generation code by xTaskCreate() is output to Processing\_Before\_Start\_Kernel() in src/frtos\_startup/freertos\_start.c . Include rtos/r\_ble\_rtos.h that is header for the BLE task control in freertos\_start.c and change the parameters (Table **6.1**) for main task generation as Figure **6-3**.

Table 6.1	:	Main	task	generation	parameters
-----------	---	------	------	------------	------------

Parameter	Description
Stack size	1024
	Stack size is multiplied by 4 (4096bytes).
Priority	6 (higher than the subtask)
Task handle	g_ble_task_hdl

<pre>#include "rtos/r_ble_rtos.h"</pre>
/* some code is omitted.*/
<pre>/*********** task creation *******************/ /* Main task. */ ret = xTaskCreate(main_task, "MAIN_TASK", 1024, NULL, 6, &amp;g_ble_task_hdl); if (pdPASS != ret) {     Stack Size Priority Task handle } </pre>
<pre>while(1) {     /* Failed! Task cannot be created. */   } }</pre>

Figure 6-3 : BLE task generation code as main task

#### (2) Call to app\_main()

Change main\_task() in src/"{Project Name}".c output by code generation to call app\_main().

```
#include "FreeRTOS.h"
#include "task.h"
void app_main(void);
void main_task(void *pvParameters)
{
    app_main();
    vTaskDelete(NULL);
}
```

Figure 6-4 : Calling app\_main() in main\_task()



#### (3) Changes to app\_main.c

Include "FreeRTOS.h" and "task.h" to call FreeRTOS API and change the main loop in app\_main() to pass control to other tasks when the Bluetooth LE Protocol Stack is idle.

```
#include "FreeRTOS.h"
#include "task.h"
/* some code is omitted. */
/* main loop */
while (1)
{
    /* some code is omitted. */
    /* process Event */
    R_BLE_Execute();
    if(0 != R_BLE_IsTaskFree())
    {
        ulTaskNotifyTake(pdFALSE, portMAX_DELAY);
    }
}
```



#### (4) Subtask generation

If subtask is generated by "4.5.5(2) Subtask Declaration", call Object\_init\_manual() from the application code such as "{Project Name}.c" or app\_main.c.

```
void app_main(void)
{
    /* some code is omitted. */
    extern void Object_init_manual (void);
    Object_init_manual();
    /* main loop */
    while (1)
    {
        /* some code is omitted. */
    }
}
```

Figure 6-6 : Subtask generation



#### 6.2.2 Generation as subtask

This section describes how to generate the BLE task as subtask according to "4.5.5(2) Subtask Declaration".

#### (1) Register BLE task

If the BLE task is registered by FreeRTOS Object configuration, configure the parameters in Table 6.1 as Figure 6-7. The stack size is multiplied by 4 (4096 bytes).

H	leap	Estimation	Та	sks	Semap	hores	Queues	Softwar	e Tim	ers	Event Group	os	Stream Buffe	rs N	lessage	Buffers	5	
	+/-	Initialize		Task C	ode	Task N	lame	Stack Size	, _	Task	Handler		Parameter	Pric	ority	He	ap Usage	
	0	manual	~	ble_t	аѕк	ble_t	аѕк	1024		g_t	DIE_task_hdi		NULL	6		0		

Figure 6-7 : BLE task registration in FreeRTOS Object configuration

#### (2) Call to app\_main()

Change ble\_task() in src/rtos\_skeleton/ble\_task.c output by code generation to call app\_main() and include "FreeRTOS.h" and "task.h" to call FreeRTOS API. An example of ble\_task.c is shown below.

```
/* Start user code for import. Do not edit comment generated here */
#include "FreeRTOS.h"
#include "task.h"
void app_main(void);
/* End user code. Do not edit comment generated here */
void ble_task(void * pvParameters)
{
    /* Start user code for function. Do not edit comment generated here */
    app_main();
    vTaskDelete(NULL);
    /* End user code. Do not edit comment generated here */
}
```

Figure 6-8 : Sample of ble\_task.c

#### (3) Changes to app\_main.c

Refer to "6.2.1(3) Changes to app\_main.c".



#### (4) Include the BLE task control header and delete task handle

Delete the BLE task handle definition in src/frtos\_startup/freertos\_object\_init.c and include the BLE task control header rtos/r\_ble\_rtos.h.

```
#include "FreeRTOS.h"
#include "freertos_start.h"
#include "../frtos_skeleton/task_function.h"
#include "task.h"
/* Include the BLE task control header */
#include "rtos/r_ble_rtos.h"
/* Some code is omitted. */
void Kernel_Object_init (void);
void Object_init_manual (void);
BaseType_t ret;
/* Delete the BLE task handle definition. */
/* TaskHandle_t g_ble_task_hdl = NULL; */
```

Figure 6-9 : Changes to freertos\_object\_init.c

#### (5) Subtask generation

Call Object\_init\_manual() from the main task code such as "{Project Name}.c" . An example of starting subtasks and do nothing in main loop is shown below.

```
#include "FreeRTOS.h"
#include "task.h"
void main_task(void *pvParameters)
{
    /* Create all other application tasks here */
    void Object_init_manual (void);
    Object_init_manual();
    while(1)
    {
    }
}
```

Figure 6-10 : Subtask generation



## 6.3 BLE GATT App task generation

This section describes how to generate according to "4.5.5(2) Subtask Declaration".

#### (1) Register BLE GATT App task

Figure 6-11 shows a sample of BLE GATT App task registration in FreeRTOS Object configuration.

H	leap E	stimation	Tasks	Semaphores	Queues S	Softwa	re Timers	Event	Groups	Stream	Buffers	Messag	e Buffers		
	+/-	Initialize		Task Code	Task Nam	e	Stack Siz	e	Task Ha	ndler	Parame	ter	Priority	Heap Usage	
	•	manual	~	bas_task	bas_task		256		g_bas_	task	conn_	hdl	4	0	
	•	manual	~	lss_task	lss_task		256		g_lss_t	ask	conn_	hdl	4	0	
	0														

Figure 6-11 : BLE GATT App task registration in FreeRTOS Object configuration

#### (2) Header files included in BLE GATT App task file

BLE GATT App task is implemented in src/rtos\_skeleton/"Task Code".c output by code generation. BLE GATT App task needs the following header files.

- GATT service API header .../Config\_BLE\_Profile/r\_ble\_[GATT service abbreviation].h
- BLE task control header rtos/r\_ble\_rtos.h
- FreeRTOS API header FreeRTOS.h and task.h
- Subtask header .../ frtos\_skeleton/task\_function.h

```
/* FreeRTOS API header */
#include "FreeRTOS.h"
#include "task.h"
/* GATT service API header */
#include "../Config_BLE_Profile/r_ble_bas.h"
/* BLE task control header */
#include "rtos/r_ble_rtos.h"
```

Figure 6-12 : Headers included in BLE GATT App task

The event definitions for task communication to other subtask are implemented in the subtask header (task\_function.h). A sample of subtask header is shown below.

```
/* some code is omitted. */
/* bas */
#define BAS_WAIT_EN_CCCD (0x0001)
/* lss */
#define LSS_NOTIF_DIS_CCCD (0x0000)
#define LSS_WAIT_EN_CCCD (0x0001)
#define LSS_WAIT_DIS_CCCD (0x0002)
#define LSS_WAIT_PUSH_SW (0x0004)
#define LSS_WAIT_WR_BLINK (0x0008)
```

Figure 6-13 : Sample of subtask header



#### (3) Edit the task generation API (Object\_init\_manual)

If the parameter is passed to the task function when the task is generated according to (1), prepare a static variable for storing the parameter and specify the static variable in xTaskCreate. An example of task generation API passing a connection handle when establishing a connection is shown below.

```
static uint16_t gs_conn_hdl;
/* some code is omitted. */
void Object_init_manual (uint16_t conn_hdl)
{
    gs_conn_hdl = conn_hdl;
    /*********** task creation ***********************/
    ret = xTaskCreate(bas_task, "bas_task", 256, &gs_conn_hdl, 4, &g_bas_task);
    /* some code is omitted. */
    ret = xTaskCreate(lss_task, "lss_task", 256, &gs_conn_hdl, 4, &g_lss_task);
    /* some code is omitted. */
} /* End of function Object_init_manual()*/
```



## (4) Call to the task generation API (Object\_init\_manual)

Refer to "6.2.1(4) Subtask generation".



## 6.4 Wake up BLE task from another task

If other task wakes the BLE task up, include the BLE task control header rtos/r\_ble\_rtos.h and call R\_BLE\_RTOS\_WakeTask(). An example of task switching from BLE GATT App task (LED and Switch service) to BLE task is shown below.

```
/* Include BLE task control header */
#include "rtos/r ble rtos.h"
/* some code is omitted. */
static uint16_t gs_conn_hdl;
void lss_task(void * pvParameters)
{
    uint8_t push_state;
    ble_status_t retval;
    /* some code is omitted. */
    retval = R_BLE_LSS_NotifySwitchState(gs_conn_hdl, &push_state);
    if(BLE_SUCCESS == retval)
    {
        R_BLE_CLI_Printf("R_BLE_LSS_NotifySwitchState \n");
        R_BLE_RTOS_WakeTask();
    ł
    /* some code is omitted. */
}
```

Figure 6-15 : Sample of task switching from BLE GATT App task to BLE task



#### 6.5 Wake up BLE task from interrupt other than RF

The BLE FIT module provides mechanisms that task is waked up from registered function by the SCI interrupt using the console function or the IRQ interrupt using LED & SW.

#### 6.5.1 Register task waked up by command input

If a task is waked up with the SCI interrupt by command input, included rtos/r\_ble\_rtos.h and call R\_BLE\_CLI\_RegisterEventCb() in application initialization to register a callback for waking up the task. An example of registering a function that wakes the BLE task up from ISR to communicate with Bluetooth LE by command input is shown below.

```
#include "rtos/r_ble_rtos.h"
/* some code is omitted */
void app_main(void)
{
    /* Initialize BLE */
    R_BLE_Open();
    /* some code is omitted */
    /* Configure Command Line */
    R_BLE_CLI_Init();
    R_BLE_CLI_RegisterCmds(gsp_cmds, ARRAY_SIZE(gsp_cmds));
    /* Register task waked up by command input */
    R_BLE_CLI_RegisterEventCb(R_BLE_RTOS_WakeTaskFromIsr);
    R_BLE_CMD_SetResetCb(ble_host_stack_init);
```

Figure 6-16 : Sample of registering a task waked up by command input

#### 6.5.2 Wake BLE task up using LED & SW

If a task is waked up with the IRQ interrupt by pressing SW, included rtos/r\_ble\_rtos.h and call R\_BLE\_CLI\_RegisterEventCb() in application initialization to register a callback for waking up the task. An example of registering a function that wakes the BLE task up from ISR to communicate with Bluetooth LE by pressing SW is shown below.

```
#include "rtos/r_ble_rtos.h"
/* some code is omitted */
void app_main(void)
{
    /* Initialize BLE */
    R_BLE_Open();
    /* Configure the board */
    R_BLE_BOARD_Init();
    R_BLE_BOARD_RegisterSwitchCb(BLE_BOARD_SW2, sw_cb);
    /* Register task waked up by pressing SW */
    R_BLE_BOARD_RegisterSwitchEventCb(R_BLE_RTOS_WakeTaskFromIsr);
```

Figure 6-17 : Sample of registering a task waked up by pressing SW



## 7. Tools

The BLE FIT module package available for download from Renesas Bluetooth Low Energy Protocol Stack for RX family Web page (<u>https://www.renesas.com/software-tool/bluetooth-low-energy-protocol-stack-rx-family</u>) provides the tools for the HCI mode firmware. For more details centering the HCI mode, see "Bluetooth Low Energy Protocol Stack Basic Package: User's Manual (R01UW0205), 6. HCI Mode".

r01an4860xxXXXX-rx23w-ble-fit.zip	XXXX: Revision number
FITDemos\	Sample application folder
FITModules\	FIT module folder
utilities\	Utility folder
BDAddrWriter.zip	Public BD address writing tool for HCI mode
CLVALTune.zip	Operation tool for calibration for HCI mode
I	

Figure 7-1 : HCI mode tool

## 7.1 BDAddrWriter

BDAddrWriter configures BD\_ADDR in the board for HCI mode. For more details, see "Bluetooth Low Energy Protocol Stack Basic Package: User's Manual (R01UW0205), 5.6.4.2 Write to data area using BDAddrWriter tool".

## 7.2 CLVALTune

CLVALTune is a tool used for calibration operation. Refer to "RX23W Group Tuning procedure of Bluetooth dedicated clock frequency(R01AN4762)" for details.



## 8. Demo Projects

The applications shown in Table 8.1 are provided as the demo projects.

The demo projects include a main function that uses the BLE FIT module and the related FIT modules (e.g.,  $r_bsp$ ).

Demo Project	Description
ble_demo_rsskrx23w_profile_client	GATT Client demo application for RSSK.
ble_demo_rsskrx23w_profile_server	GATT Server demo application for RSSK.
ble_demo_rsskrx23w_uart_hci	HCI mode demo application for RSSK.
ble_demo_tbrx23w_profile_client	GATT Client demo application for Target Board for RX23W.
ble_demo_tbrx23w_profile_server	GATT Server demo application for Target Board for RX23W.
ble_demo_tbrx23w_uart_hci	HCI mode demo application for Target Board for RX23W.
ble_demo_tbrx23wmodule_profile_client	GATT Client demo application for Target Board for RX23W module.
ble_demo_tbrx23wmodule_profile_server	GATT Server demo application for Target Board for RX23W module.
ble_demo_tbrx23wmodule_uart_hci	HCI mode demo application for Target Board for RX23W module.
ble_demo_tbrx23w_FreeRTOS_multi_services	GATT Server demo application on Renesas FreeRTOS for Target Board.

Tahle	8 1	Demo	Proi	orte
Iable	0.1	Denio	FIU	ecis

The above FITDemos use the same version of the r\_ble\_rx23w as this document. The version of other FIT modules is the same as the one included in RX Driver Package v1.36 (<u>https://www.renesas.com/software-tool/rx-driver-package</u>). The FITDemo projects use QE for BLE V1.6.0.

The GATT server and the GATT Client demo projects support the command line interface. By connecting the board and PC with serial interface, it allows to input commands and output debug messages on a serial terminal. The terminal configuration is shown in Table **2.6**.



## 8.1 GATT Server demo project

The GATT Server demo works as below.

- After starting, it starts advertising and waits for a command.
- By scanning from a remote device, it is detected by the "RBLE-DEV" device name.

III docomo 3G	11:15	• 100% 💷 ا
) <mark>I</mark> (	GATT Browser	Menu
RBLE-DEV	-7D87-EF87-D5C35288366E	
<no name<="" td=""><td>&gt; A-9471-EC9B-CD7E57D8E1C8</td><td></td></no>	> A-9471-EC9B-CD7E57D8E1C8	
<no name<="" td=""><td>&gt; -1E53-BDAE-506CDDC97CB1</td><td></td></no>	> -1E53-BDAE-506CDDC97CB1	

Figure 8-1 : Scan result example

- When connected, it stops advertising.
- The followings are detected by searching GATT services from a remote device.
  - ▶ LED Switch Service (LSS, UUID: 58831926-5F05-4267-AB01-B4968E8EFCE0)
  - Switch State Characteristic (UUID: 58837F57-5F05-4267-AB01-B4968E8EFCE0)
  - > LED Blink Rate Characteristic (UUID: 5883C32F-5F05-4267-AB01-B4968E8EFCE0)

III docomo 3G	11:16	🎱 100% 🥅
🗸 Back	Services	Disconnect
RBLE-DEV		
Device UUID : 455D19 Connection Status : (	90-1CE8-7D87-EF87-D5C: Connected	35288366E
Custom Service UUID : 58837F57-5F0 Properties : Notify	58831926-5F05-4267-, 5-4267-AB01-B49	AB01-B4968E8EFCE0
5883C32F-5F0 Properties : Read	5-4267-AB01-B49 <sup>Write</sup>	68E8EFCE0

Figure 8-2 : Service Discovery result



 If the LED Switch Service second parameter in the gs\_gatt\_service variable in the gatt\_db.c is set to BLE\_GATT\_DB\_SER\_SECURITY\_UNAUTH | BLE\_GATT\_DB\_SER\_SECURITY\_ENC, the GATT Server requests pairing to access to the Characteristic in the LED Switch Service. If 0 is set, pairing is not necessary.

```
static const st_ble_gatts_db_serv_cfg_t gs_gatt_service[] =
{
    /* some code is omitted. */
    /* LED Switch */
    {
        /* Num of Services */
        {
            1,
        },
/* Description */
        BLE_GATT_DB_SER_SECURITY_UNAUTH | BLE_GATT_DB_SER_SECURITY_ENC,
                                                                                Pairing is necessary.
           Service Start Handle
        0x0010,
        /* Service End Handle */
        0x0015,
        /* Characteristic Start Index */
        6,
        /* Characteristic End Index */
        7,
    },
};
```



Figure 8-3 : The security setting of the access to the LED Switch Service.

- After enabling the notification in the Switch State characteristic, the notification packet is sent to the remote device by pushing the SW1 on the board.
- By writing a number to the LED Blink Rate characteristic, the LED blinks at the number x 100ms interval. The LED turns off by writing zero to the characteristic.
- When disconnected, it restarts advertising.



Figure 8-4 shows usage example for the GATT Server demo project and a remote device.



Figure 8-4 : Usage example for the GATT Server demo project and a remote device



## 8.2 GATT Client demo project

The GATT Client demo works as below.

- After starting, it waits for a command.
- If you would like to confirm the presence of the GATT Server demo project, execute the "gap scan" command by your terminal software and the advertising packet are received. Refer to "Bluetooth Low Energy Protocol Stack Basic Package: User's Manual (R01UW0205), 5.1.1.1 (2) Scan command" for the details of the "gap scan" command. After receiving the advertising, stop scan by press CTRL + C on the terminal software.
- Execute the "gap conn" command by the terminal software to connect with the GATT Server demo project. Refer to "Bluetooth Low Energy Protocol Stack Basic Package: User's Manual (R01UW0205), 5.1.1.1 (3) Connection command" for the details of the "gap conn" command.
- When connected, packet data length exchange is done.
- When the packet data length changed, it sends the MTU exchange request.
- Receiving the MTU exchange response, it starts searching GATT services in the remote device.
- If the GATT Server demo project is set to request pairing (BLE\_GATT\_DB\_SER\_SECURITY\_UNAUTH | BLE\_GATT\_DB\_SER\_SECURITY\_ENC: Figure 8-3), pairing needs to be done before to the LED Switch client command execution. Input the "gap auth" command to start pairing. Refer to "Bluetooth Low Energy Protocol Stack Basic Package: User's Manual (R01UW0205), 5.1.1.1 (9) Authentication command" for the details of the "gap auth" command.
- After the completion of the above procedure, it allows to use the following LED Switch client commands.
  - Enable / Disable Notification lsc set\_switch\_state\_ntf [connection handle] [0(disabled) or 1(enabled)]
  - Write the LED blink rate lsc write\_led\_blink\_rate [connection handle] [blink\_rate] The LED blinks at the [blink\_rate ] x 100ms. Valid range is 0-255. The LED turns off by writing 0.
- Execute the "gap disconn" command by the terminal software to disconnect the GATT Server demo project. Refer to "Bluetooth Low Energy Protocol Stack Basic Package: User's Manual (R01UW0205), 5.1.1.1 (4) Disconnection command" for the details of the "gap disconn" command.



Figure 8-5 shows usage example for the GATT Client demo project and the GATT Server demo project.



Figure 8-5 : Usage example for the GATT Client demo project and the GATT Server demo project



## 8.3 Renesas FreeRTOS BLE demo project

The GATT Server demo works as below. This project consists of Battery Service task (BAS task) and LED & SW Service task (LSS task).

- After starting, it starts advertising and waits for a command.
- By scanning from a remote device, it is detected by the "RBLE-DEV" device name.

•••• docomo 3G	11:15	• 100% ا
° <b>I</b> C	GATT Browser	Menu
RBLE-DE	EV CE8-7D87-EF87-D5C35288366E	
<no nam<="" td=""><td>1<b>C&gt;</b> EAA-9471-EC9B-CD7E57D8E1C8</td><td></td></no>	1 <b>C&gt;</b> EAA-9471-EC9B-CD7E57D8E1C8	
<no nam<="" td=""><td>1<b>C&gt;</b> 196-1E53-BDAE-506CDDC97CB1</td><td></td></no>	1 <b>C&gt;</b> 196-1E53-BDAE-506CDDC97CB1	

Figure 8-6 : Scan result example

- When connected, it stops advertising.
- The followings are detected by searching GATT services from a remote device.
  - Battery Service (BAS, UUID: 180F)
  - Battery Level Characteristic (UUID: 2A19)
  - LED Switch Service (LSS, UUID: 58831926-5F05-4267-AB01-B4968E8EFCE0)
  - Switch State Characteristic (UUID: 58837F57-5F05-4267-AB01-B4968E8EFCE0)
  - LED Blink Rate Characteristic (UUID: 5883C32F-5F05-4267-AB01-B4968E8EFCE0)

Il docomo 3G	2:11	💭 87% 🕖
🗸 Back	Services	Disconnect
USER_TASK		
Device UUID : 100467	E5-70E5-9A22-5D57-6660	5AD07614
Connection Status : (	Connected	
Battery Service UUI	D: 180F	
Battery Level		
Properties : Read	Notify	
Custom Service UUID :	: 58831926-5F05-4267-/	AB01-B4968E8EFCE0
58837F57-5F0	5-4267-AB01-B49	68E8EFCE0
Properties : Notify		
5883C32F-5F0	5-4267-AB01-B49	68E8EFCE0
Properties : Read	Write	· · · · · · · · · · · · · · · · · · ·

Figure 8-7 : Service Discovery result

- After enabling the notification in the Battery Level characteristic, the notification packet including a dummy Battery Level is sent to the remote device from RX23W every 1 second.
- After enabling the notification in the Switch State characteristic, the notification packet is sent to the remote device by pushing the SW1 on the board.
- By writing a number to the LED Blink Rate characteristic, the LED blinks at the number x 100ms interval. The LED turns off by writing zero to the characteristic.
- When disconnected, it restarts advertising.



Figure 8-8 shows usage example for the Renesas FreeRTOS BLE demo project and a remote device.



Figure 8-8 : Usage example for the Renesas FreeRTOS BLE demo project and a remote device



## 8.4 HCI mode demo project

The HCI mode demo project waits for an HCI command after starting. Input an HCI command for the RF characteristic evaluation or connect to BTTS (Bluetooth Test Tool Suite: R01AN4554).



## 8.5 Operation confirmation on CS+

Using the Renesas common project file (rcpc file) in demo project, you can do an operation confirmation of the demo projects on CS+. The version of CS+ used for operation confirmation is V8.05. Open the demo project on CS+ according to the following procedure.

- 1. Run CS+.
- 2. Click "File >> Open File", set the filename extension to "Project File foe MCU Simulator Online/ e<sup>2</sup> studio (\*.rcpc) and select the demo project rcpc file.

🚳 Open File						Х
	<pre> src → git_repo_</pre>	SDK_proto3 → proje	ects > ble_demo_tbrx23w_profile_clien	t ∨ Ö	Search ble_demo_tbrx23	Bw_pr 🔎
Organize 👻 New	folder					
Documents	^	Name	^	Date modified	Туре	Size
👆 Downloads		src		07/06/2021 5:28 PM	File folder	
b Music		🕼 ble_demo_t	tbrx23w_profile_client.rcpc	31/05/2021 10:11 AM	RCPC File	33
Pictures						
😽 Videos						
🏭 Windows (C:)						
💣 Network						
	~	, «				,
F	ile name: ble_den	no_tbrx23w_profile_c	lient.rcpc	~	Project File for MCU Sir	mulator i 🗸
					Open	Cancel

Figure 8-9 : Open the demo project rcpc file

3. Press "OK" on "Project Convert Settings" screen.



Figure 8-10 : Project Convert Settings



4. After the project conversion, you can do an operation confirmation on CS+. For the CS+ operation, refer to the following User's Manual. <u>https://www.renesas.com/software-tool/cs</u>

🕹 🛍 🔊 (~) 🏭 🌉 🏨 🔍 🔍 100% 💌 🚺	🔐 💱 HardwareDebug 🔹 🖌 🤅 🖓 🦄 🐂 🗐 💿 🕞 🗠 🐼 🧐 🖘 🐇
🐺 Project Tree 📮 🗶	Property 💌 🗙
S 2 3 2 3	
Rible demo thr/23w profile client (Project)	ble_demo_tbrx23w_profile_client Property
BSE523W84vNG (Microcontroller)	File
Smart Configurator (Design Tool)	Absolute path C:\src\rdit repo SDK proto3\projects\ble_demo_tbrx23w
CC-RX (Build Tool)	License
RX Simulator (Debug Tool)	> Notes
Program Analyzer (Analyze Tool)	
🖃 👘 File	
src	
ProjectConvertInformation_ble_demo_tbrx23w_profile_client.txt	
	File name
	This is the name of the file to which the information of this main project is to be saved.
	Project
	Output 7 🗙
	(EOF)
	Anriessages
< >	🔜 Output 😻 Smart Browser 🛅 Error List
	-X DISCONNECT

Figure 8-11 : Demo project on CS+



## 8.6 Importing the demo project

The sample application provided with this document may be imported into  $e^2$  studio using the steps in this section.

- (1) Create a RX23W project including the BLE FIT module(r\_ble\_rx23w), referring to 4.1 Create a new project and 4.4 Adding FIT modules. Or prepare an existing RX23W project including the BLE FIT module.
- (2) Select "Components" tab, right-click "r\_ble\_rx23w" and click "Download and import sample projects".

Software component config	guration		🐻 🖨
Components $ _{\mathbb{Z}}^{a} \models + \rightarrow$	Configure		
type filter text	Property <ul> <li></li></ul>	Value	^
Gpo mor cont	# Type of BLE protocol stack library	All features	
✓ → Startup	# Initial Public Address.	{0xFF,0xFF,0xFF,0x50,0x90,0x74}	
✓ Generic	# Initial Static Address.	{0xFF,0xFF,0xFF,0xFF,0xFF,0xFF}	
er_bsp	# Maximum number of connections.	7	
✓ → Drivers	# Maximum connection data length.	251	
<ul> <li>Low power consumption</li> </ul>	# Maximum advertising data length.	1650	
r_lpc_rx	# Maximum advertising set number.	4	
✓ → Timers	# Maximum periodic sync set number.	2	
r_cmt_rx	# Connection event start notify.	Disable	
✓ → Middleware	# Connection event close notify.	Disable	~
✓ Generic	<		>
r_ble_rx23w	Change version		^
*	Remove Reset to default		~
	Download and import sample projects		
Overview Board Clocks Components	s Pins Interrupts		

Figure 8-12 : Download and import sample projects (1)

(3) After Smart Browser displays the BLE FIT module application note (Title: RX23W Group BLE Module Firmware Integration Technology Application Note, Document No: R01AN4860EJYYYY (YYYY: version)), right-click this and select "Sample Code (import projects).

🚺 Memory 🔅 Debug 🌸 Smart Browser 🔀			송 양 북	a) 🔂 💥 🧃	k 🗸 🗉 🗆
Device: R5F523W8A(RX23W)	Last updated: 2020/04/07	at 17:13:24 JST			
Context Help User's Manual Technical Update Application Notes Tool News Notification	ons				
152 matches					
Title	Document No.	Rev.	Issue Date	Sample Code	Remarks
RX23W Group BLE Module Firmware Integration Technology Application Note	R01AN4860EJ0110	Rev.1.10	2019/12/26	available	
Bluetooth® Low Energy microcomputer Design Guidelines for a Pattern Antenn	Open		2019/12/26	-	
RX23W Bluetooth Test Tool Suite operating instructions Application Note	Sample Code (download	)	2019/12/20	available	
Renesas Flash Programmer Usage from the Command Line	Sample Code (import pro	ojects)	2019/12/19	-	
RX Family RX Driver Package Ver.1.23	Property		2019/12/16	available	
RX Family QE for BLE[RX] R_BLE Script sample and dedicated program Application	R01AN4872EJ0100	Rev.1.00	2019/12/04	available	

Figure 8-13 : Download and import sample projects (2)



(4) Input the location where r01an4860xxYYYY-rx23w-ble-fit.zip (YYYY: version) is downloaded. After the download has been completed, "Select import package" window is displayed. Select the demo project.

e <sup>2</sup> Select import	package	$\times$
Import package		~
	r01an4860xx0110-rx23w-ble-fit¥FITDemos¥ble_demo_rsskrx23w_profile_client.zip	
	r01an4860xx0110-rx23w-ble-fit¥FITDemos¥ble_demo_rsskrx23w_profile_server.zip	
	r01an4860xx0110-rx23w-ble-fit¥FITDemos¥ble_demo_rsskrx23w_uart_hci.zip	
	r01an4860xx0110-rx23w-ble-fit¥FITDemos¥ble_demo_tbrx23w_profile_client.zip	n
	r01an4860xx0110-rx23w-ble-fit¥FITDemos¥ble_demo_tbrx23w_profile_server.zip	
	r01an4860xx0110-rx23w-ble-fit¥FITDemos¥ble_demo_tbrx23w_uart_hci.zip	

Figure 8-14 : Download and import sample projects (3)

(5) When "Finish" button on "Import" window has been pressed, the demo project is imported.

e <sup>2</sup> Import					×
Import Projects					
Select a directory to sea	ch for existing Eclipse projects.				
○ Select root directory:		~	B	rowse	
• Select archive file:	/-ble-fit¥FITDemos¥ble_demo_tbrx23w_profile_serve	r.zip ~	В	rowse	
Projects:					
✓ ble_demo_tbrx23w	_profile_server (ble_demo_tbrx23w_profile_server/)		S	elect All	
			De	select Al	I
			I	Refresh	
Options					
Search for nested pro	jects				
Copy projects into we	prkspace				
Hide projects that alr	eady exist in the workspace				
Working sets					
Add project to work	ing sets		N	ew	
Working sets:		~	Sel	ect	
2	Finish			Cancol	
	Finish			Cancer	

Figure 8-15 : Download and import sample projects (4)



## 9. Appendices

## 9.1 Confirmed Operation Environment

This section describes confirmed operation environment for the BLE FIT module.

Table 9.1 : Confirmed Operation B	Environment (Rev.2.50)
-----------------------------------	------------------------

Item	Contents
Integrated development	Renesas Electronics e <sup>2</sup> studio 2022-10 (64bit)
environment	IAR Embedded Workbench for Renesas RX 4.20.1
C compiler	Renesas Electronics C/C++ Compiler Package for RX Family V2.08.00
	IAR C/C++ Compiler for Renesas RX version 4.20.1
	Compiler option: The default settings of the integrated development environment.
Endian	little endian
Revision of the module	Rev 2.50
RTOS	FreeRTOS (kernel only) version 10.4.3-rx-1.0.6
Board used	Target Board for RX23W (RTK5RX23W0C00000BJ)
	Target Board for RX23W module (RTK5RX23W0C01000BJ)
	Renesas Solution Starter Kit for KX23VV (RTK5523VV8AC00001BJ)



## 9.2 Troubleshooting

- (1) Q: I have added the FIT module to the project and built it. Then I got the error: Could not open source file "platform.h".
  - A: The FIT module may not be added to the project properly. When using this FIT module, the board support package FIT module (BSP module) must also be added to the project. Refer to the application note "Board Support Package Module Using Firmware Integration Technology (R01AN1685)".
- (2) Q: I have added the FIT module to the project and built it. Then I got the error: This MCU is not supported by the current r\_ble\_rx23w module.
  - A: The FIT module you added may not support the target device chosen in your project. Check the supported devices of added FIT modules.
- (3) Q: I have added the FIT module to the project and built it. Then I got an error for when the configuration setting is wrong.
  - A: The setting in the file "r\_ble\_rx23w\_config.h" may be wrong. Check the file "r\_ble\_rx23w\_config.h". If there is a wrong setting, set the correct value for that. Refer to 2.6, Configuration Overview for details.

(4) Q: I have added the FIT module to the project and built it. Then I got an error: Could not open source file.

A: The file path may be too long. Shorten the file path.



## **Revision History**

Rev.	Date	Description			
		Page	Summary		
1.00	Aug.23.2019	—	First edition issued.		
1.01	Oct.31.2019	20	Move the description of the command to "Bluetooth <sup>®</sup> Low Energy Protocol Stack Basic Package: User's Manual (R01UW0205) ".		
		26	Update "4. How to use".		
		58	Add how to use the demo application to "7. Demo Project".		
		Library	Add Mesh support.		
		Library	Support the pairing when changing the BD_ADDR by R_BLE_VS_SetBdAddr() with RBLE_VS_ADDR_AREA_REG(0x00).		
1.10	Dec.26.2019	—	Supported the following compilers. - IAR C/C++ Compiler for Renesas RX		
		Library	<ul> <li>Fixed the issue that pairing fails with the max bonding information after restart.</li> <li>Fixed the issue that the BLE_GAP_EVENT_PAIRING_COMP event is not notified where R_BLE_GAP_StartPairing() is called after the local device deleted the bonding information.</li> <li>Fixed the issue where a connection with the second and following paired remote device which of the address was resolved is not established after restart.</li> <li>Fixed the issue that the bonding information is not deleted where the remote parameter of R_BLE_GAP_DeleteBondInfo() is specified as BLE_GAP_SEC_DEL_REM_NOT_CONN(0x02).</li> <li>Changed the host stack to return error when adding or deleting a device from Resolving List / White List / Periodic Advertiser List in clearing the list.</li> <li>Fixed an issue where the link is disconnected in high throughput</li> </ul>		
			<ul> <li>communication when a request from the remote device is accepted.</li> <li>Fixed the RSSI value error correction.</li> <li>Changed the MTU exchange request minimum size to 23 bytes.</li> <li>Fixed an issue that local device using the balance library couldn't send a connection request in adverting.</li> </ul>		
		Program	<ul> <li>Added the BSP macro as the BLE interrupt function declaration and the section.</li> <li>Changed the Abstraction advertising API to set the random own address type.</li> <li>Moved the response for a connection parameter update request from the command line to application layer.</li> <li>Disabled the code in the cmd directory when BLE_CFG_CMD_LINE_EN is set to 0.</li> <li>Fixed the memory access violation where BLE_CFG_RF_ADV_SET_MAX is other than 4.</li> <li>Added the scan process where the ad type is specified as 0.</li> <li>Fixed the other FIT modules dependencies and changed the following configuration option default value.</li> <li>BLE_CFG_CMD_LINE_EN (r_sci_rx, r_byteq_rx))</li> <li>BLE_CFG_BOARD_LED_SW_EN (r_gpio_rx, r_irq_rx)</li> <li>BLE_CFG_DEV_DATA_DF_BLOCK (r_flash_rx)</li> </ul>		



			- Added the following configuration option.
			BLE_CFG_ABS_API_EN (enable/disable abstraction API)
			BLE_CFG_SOFT_TIMER_EN (enable/disable software timer)
			BLE_CFG_MCU_LPC_EN (enable/disable MCU low power
			consumption control)
			BLE_CFG_HCI_MODE_EN (enable/disable HCI mode)
2.00	Jul.27.2020	8, 44	- Added the description that the CMT code does not need to be
			changed if the version is 4.50 or later.
		14	- Added the notice about BLE_CFG_NUM_BOND.
		17	- Move the description of the features in Table 2.2 to the User's
			Manual (R01UW0205).
		30	- Added 4.2 Confirm BSP version.
		32	- Modified the SCI FIT Module name.
			r_sic_rx → r_sci_rx
		33	- Added the description about a method for displaying all FIT
			modules.
		36-37	- Added the description about the SCI FIT Module configuration when
			the command line is used.
		39-40	- Added the initial value for the customer board to Table 4.1.
			- Added the IRQ configuration option settings.
		47	- Modified the code changes in "4.6.3 Add application code".
		55-68	- Added the description about the optimization to "4.9 Create a project
			on the IAR development".
		69	- Move the contents in "5. Application Guide" to the Application.
		81	- Added the HCI mode tool location to "7. Tools".
			- Added the reference to the User's Manual (R01UW0205) to "7.1
			BDAddrWriter".
		86-87	- Modified the description about the GATT Client demo project and
			Figure 8.5.
		91-92	- Modified "8.5 Importing the demo project" to download and import
			from the Smart Configurator.
		94	- Delete the description about a method for adding FIT modules with
			the FIT configurator from 9.2 (1) answer.
			- Change the code sample style to the e <sup>2</sup> studio editor style.
			- Corrected the API specification name.
			r ble spec.chm> r ble api spec.chm
		Library	- Added Scan Channel Map API.
			- Fixed the RF event notification in Balance and Compact libraries.
			- Fixed the transmission power change feature by
			R_BLE_VS_SetTxPower().
			- Fixed the long characteristic write completion notification.
			- Added the measures of the vulnerability to invalid packets.
			- Fixed the problem regarding slave latency.
			- Fixed the problem regarding bonding
		Program	- Added the support for Renesas FreeRTOS
			- Modified the condition for moving to the software standby state
			- Fixed the CCCD confirmation for GATT Notification and Indication
1	1	1	i issue the every communication for over i reduited tori and indication.



2 10	Dec 2/ 2020	6	- Modified "Figure 1.1: Software structure"
2.10	Dec.24.2020	Library	Fixed the following huge
		Library	<ul> <li>The RF wake-up process cannot be conducted normally, if the API execution process in R_BLE_Execute() and a RF wake-up interrupt compete in limited timing (a few cycles).</li> </ul>
			<ul> <li>Scan cannot continue depending on the timing of repeated advertising turning on and off.</li> </ul>
			<ul> <li>Software timer fails to expire after a specified interval if R_BLE_TIMER_UpdateTimeout() was continuously called.</li> </ul>
			- A Central (Master) not using RPA fails to connect to a Peripheral (Slave) using RPA.
			- A GATT server returns "Insufficient Authentication Error" during service discovery, if check "Encryption" of "Included Service" on the QE for BLE.
		Program	Modified the followings.
			<ul> <li>The scan and create connection abstraction APIs can specify the local device address type.</li> </ul>
			- The software timer callback function wakes-up the BLE task when a project includes Renesas FreeRTOS
			<ul> <li>The updating process of the bonding information in the security data management.</li> </ul>
			Added the following options to the Command Line.
			<ul> <li>Local device address type option for adv, scan, conn command.</li> </ul>
			- Use White List option for adv, scan, conn command.
			- Remove an advertising set option for adv command.
			- Delete an entry in the Resolving List for priv command.
2.11	Mar.30.2021	-	- Support RX23W module (R5F523W8CxLN, R5F523W8DxLN).
		8	- Add the version of the BSP and the GPIO FIT when selecting the RX23W module.
		12	- Add BLE_CFG_RF_CLVAL value when selecting RX23W module.
		18	- Add the note that the logger feature is not supported when using
			CCRX C++.
		20	<ul> <li>Add the note when using gap command and vs command.</li> <li>Add "2.8.7 LED and Switch control"</li> </ul>
		26	<ul> <li>Add the note that using e<sup>2</sup>studio 2021-01 or later when creating a project for RX23W module.</li> </ul>
		27	- Add the RX23W module type information to Figure 4.4.
		30	- Add the BSP version when selecting RX23W module.
		32	- Add BLE Profile Creation and r_ble_qe_utility to the components.
		33-34	<ul> <li>Add the configuration of Smart Configurator if the downloaded FIT modules are not found.</li> </ul>
		78	- Change the BLE FIT APN URL. - Add the BLE FIT package name.
		79	- Add the FITDemos for RX23W module to Table 8.1.
		92	- Add the following in Table 9.1
			<ul> <li>Add "e<sup>2</sup>studio 2021-01(64bit)" to "Integrated development environment".</li> </ul>
			<ul> <li>Add "Target Board for RX23W module" and "Renesas Solution Start Kit for RX23W to Board used".</li> </ul>



		Library	- Fixed that the changed address is reflected in advertising packet if
			static address is changed.
			- Fixed the scanner address included in a Scan Request Receive
			event (BLE_GAP_EVENT_SCAN_REQ_RECV) when using RPA
			Fixed that the DIF CAD EVENT ADV DEDT IND event is not
			- Fixed that the BLE_GAP_EVEN1_ADV_REP1_IND event is not
			only Coded PHY
			- Fixed that R BLF Execute() API does not return when the non-
			maskable interrupt status register (NMISR) is not 0x00.
		Program	- Fixed that variable length macro is not used when selecting CCRX
		C C	C++.
			- Fixed the following about Command Line.
			- Command line has come to recognize a passkey starting from 0
			as a decimal number.
			- The filter of the gap scan command has come to be able to
			include 0x00.
2.20	Jun.30.2021	88	- Added "8.5 Operation confirmation on CS+".
		Library	- Fixed for impersonation in the Passkey Entry Protocol (CVE-2020-
			26558).
2.30	Oct.15.2021	14	- Fixed the BLE_CFG_MCU_MAIN_CLK_KHZ range in Table 2.4
			when using PLL Circuit.
		52	- Fixed the build step path in 2) Register the bat file.
		Library	- Fixed a problem that main routine cannot run due to the BLEIRQ
			Interrupt continuous generation in some specific conditions when
			Eived the sheek of the main cleak range when using DLL Circuit
2 31	Apr 15 2022		Deleted descriptions about RLE Profile Creation and
2.01	Apr. 15.2022		r_ble_qe_utility.
		8	- Added the table of the Hardware Resources used by the BLE FIT.
		11	- Modified the description of BLE_CFG_RF_CONN_MAX.
		15	- Modified the description of BLE_CFG_NUM_BOND.
		22	- Modified "2.9 Flash Memory Protection" to "2.9 Device-specific
			data".
		29	- Updated the description how to create Renesas FreeRTOS project.
		70	- Added the description that R_BLE_CLI_Printf can be called by either
			task.
		79	- Fixed the URL to the BLE FIT package.
			- Fixed the zip file name in Figure 7.1.
		80	- Added the description that FITDemos use QE for BLE v1.40.
		93	- Deleted "e <sup>2</sup> studio V7.8.0 (32bit)" from Table 9.1.
		Library	- Adjusted the timing of R_BLE_Open API to stabilize the operation at
			the time of initialization.
			- Fixed R_BLE_GAP_UpdConn parameter check.
		Program	- Minor modifications in in app_lib.
0.40		10	- FII Demos support e <sup>2</sup> studio 2021-04 or later.
2.40	Jun.30.2022	10	- Added "2.3.1 Connection parameters Limitations".
		20	- Added "Do not change the Abstraction API codes.".
		Library	- Adjustment of Initialization Stability Wait Time of R_BLE_Open.
			- Stabilization of peripheral communication operation when
		Program	Connection Latericy is enabled.
		FIOGRAFII	- Fixed an issue the bonding information in the data flash cannot be



	1	1	
2.50	Dec.27.2022	-	- Profile Common moves from BLE FIT to QE for BLE (V1.6.0 or later).
		8	- Added the interrupt priority level in Table 2.1.
		10	<ul> <li>Added the description that the Supervision Timeout is shortened according to the Connection Interval to avoid the issue of not being able to transition to RF sleep mode.</li> </ul>
		50	- Added "4.6.4 Profile Common code generation" so as to prevent build error that update only BLE FIT version causes.
		55	<ul> <li>Changed the file separator character in the library selector bat file path.</li> </ul>
		Library	<ul> <li>Fixed an issue that the local device in peripheral mode can't transmit to RF sleep mode if a disconnection occurs under specific condition.</li> <li>Fixed an issue that the coding scheme (S=2/S=8) of Coded PHY could not be switched after changing to Coded PHY with R_BLE_GAP_SetPhy API.</li> <li>Fixed the GATT Database read/write procedure for variable length characteristic of which the size is larger than MTU.</li> <li>Fixed the prepare write queue handling when an error occurs.</li> <li>Fixed the registrable entry number in White List if BLE_CFG_LIB_TYPE is Balance or Compact.</li> <li>Fixed the event ID for R_BLE_GATTC_ReadCharUsingUuid.</li> <li>Fixed an issue where an unused value for local_rpa of the BLE_GAP_EVENT_CONN_IND event parameter may be notified when the White List filter is enabled with the R_BLE_GAP_CreateConn API.</li> </ul>
			<ul> <li>Fixed an issue where connection event could not be executed and BLE_GAP_EVENT_DISCONN_IND event was notified after 41 seconds when connecting with Peripheral role.</li> <li>Fixed the error code when a remote device rejects connection parameters upcate request.</li> </ul>
		Program	<ul> <li>Deleted profile_cmn, discovery from app_lib.</li> <li>Added library selector shell script.</li> <li>Fixed error bandling in security data management.</li> </ul>
		1	- Fixed error handling in security data management.



# General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

#### 2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable. 6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.).

7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

## Notice

- Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
- 2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
- 3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
- 4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
- 5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
- Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
  - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

- 7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
- 8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
- 9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
- 10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
- 12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
- This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
   Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas
- Electronics products. (Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled
- (Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
- (Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

## **Corporate Headquarters**

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan www.renesas.com

## Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

## **Contact information**

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: <u>www.renesas.com/contact/</u>.